



Ensemble Release Notes Archive

Version 2018.1
2019-09-20

Ensemble Release Notes Archive

Ensemble Version 2018.1 2019-09-20

Copyright © 2019 InterSystems Corporation

All rights reserved.



InterSystems, InterSystems Caché, InterSystems Ensemble, InterSystems HealthShare, HealthShare, InterSystems TrakCare, TrakCare, InterSystems DeepSee, and DeepSee are registered trademarks of InterSystems Corporation.



InterSystems IRIS Data Platform, InterSystems IRIS, InterSystems iKnow, Zen, and Caché Server Pages are trademarks of InterSystems Corporation.

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

About This Book	1
1 Ensemble 2017.2	3
1.1 New Features in Ensemble 2017.2	53
1.1.1 Improved XML Performance	3
1.1.2 DICOM Enhancements	3
1.1.3 XSLT2 Activity in BPL	4
1.1.4 Ensemble is now compatible with Disabling Caché Global Kill	4
1.1.5 Minor Enhancements	4
1.1.6 Caché 2017.2 Features	4
1.2 Compatibility Issues for Upgrades to Ensemble 2017.2	53
1.2.1 Do not use JMS Gateway as Message Listener	5
1.2.2 Custom RecordMap Batch Classes may need to be Modified to get Fix	5
1.2.3 BPL xslt Activity Provides More Information in Error Messages	5
1.2.4 Business Service Time to Wait Changes	5
1.2.5 Ensemble Include File Includes Definitions from %sySystem	5
1.2.6 Changes may be Required in PassthroughService Subclass	5
2 Ensemble 2017.1	7
2.1 New Features in Ensemble 2017.1	7
2.2 Compatibility Issues for Upgrades to Ensemble 2017.1	7
2.2.1 Business Process Lock Changes	8
2.2.2 XML VDoc Handling of Missing Values Changes	8
2.2.3 XML VDoc Child of Repeating Parent Changes	8
2.2.4 Ensemble Java Gateway Defaults to JDK 1.8	8
2.2.5 Ensemble DotNet and Java Gateways Support Validated Connections	8
2.2.6 Search Tables Now Handle Vertical Bars	8
3 Ensemble 2016.2	9
3.1 New Features in Ensemble 2016.2	9
3.2 Compatibility Issues for Upgrades to Ensemble 2016.2	9
3.2.1 Production Status Hover Tip May Be Stale	10
3.2.2 IsEnsembleNamespace Method Returns Information About Specified Namespace	10
3.2.3 Efficiency Improvements Changed VDoc XSD Schema Structure	10
3.2.4 Ensemble Editors Now Respect Source Control Readonly Property	10
3.2.5 Ensemble X12 Business Service Limits 999 Implementation Acknowledgement to Allowed Transaction Set Codes	11
4 Ensemble 2016.1	13
4.1 New Features in Ensemble 2016.1	13
4.2 Compatibility Issues for Upgrades to Ensemble 2016.1	13
4.2.1 EnsLib.REST.Service Handles Query Parameters Differently	14
4.2.2 Upgrading Ensemble Sets _Ensemble User to Not Expire	14
4.2.3 Business Process Termination Error Now Includes Text from Underlying Error	14
4.2.4 Ensemble Checks that Default Database Allows Write Access	14
4.2.5 Inbound TCP Adapters May Report Additional Errors on Startup	15
4.2.6 Option to Improve Restart with Suspended Messages Changes Storage Location	15
4.2.7 Improved Validation of ebXML Messages Changes Behavior	15
4.2.8 Additional Log Messages when Disconnecting Java Gateway	15

4.2.9 Email Outbound Adapter Defaults Change	15
4.2.10 FTP Failure to Delete Network File Handled Without Reprocessing File	15
4.2.11 Router Sets Message Headers Correctly After Timeout	16
4.2.12 BPL No Longer Allows Two Calls With the Same Name	16
4.2.13 TCP Counted Adapter Now Checks Incoming Length	16
4.2.14 Disabling a Business Process with a Pool Size of 0	16
4.2.15 Improved Check of Permissions for Purging Alerts	16
4.2.16 XML File and FTP Operations Write Search Table Data	17
4.2.17 Purging Completed Business Process Instances with Null TimeCreated Values	17
5 Ensemble 2015.2	19
5.1 New Features in Ensemble 2015.2	19
5.2 Compatibility Issues for Upgrades to Ensemble 2015.2	19
5.2.1 Upgrading Enterprise Message Bank Requires Rebuilding Indices	19
6 Ensemble 2015.1	21
6.1 New Features in Ensemble 2015.1	21
6.2 Compatibility Issues for Upgrades to Ensemble 2015.1	21
6.2.1 New Global and Database Used to Store Credentials Passwords	21
6.2.2 New Databases and Changes to Where Temporary Globals are Stored	22
6.2.3 JDBC SQL Returns Large Objects as Streams instead of Strings	23
6.2.4 Lookup Table Import and Export Format Changed to be Compatible with Studio	23
6.2.5 Deleting X12 Objects Deletes Child Objects	23
6.2.6 DTL foreach Error Reported Correctly	23
7 Ensemble 2014.1	25
7.1 New Features in Ensemble 2014.1	25
7.2 Compatibility Issues for Upgrades to Ensemble 2014.1	25
7.2.1 HL7 Sequence Manager Automatically Adds Index	26
7.2.2 New Security Resource to Control Exporting Deployment Packages	26
7.2.3 Ensemble HL7 Supports Long Strings	26
7.2.4 GetProductionStatus Returns Improved Status Information	26
7.2.5 Ens.Settings has New GetSettingRow Method	26
7.2.6 Ensemble CSP Pages Set UseCookies Property	26
7.2.7 Viewing EDI Raw Contents Inserts Newlines for Consistency	27
7.2.8 Improved Handling of Spaces in FTP File Names	27
7.2.9 Ensemble Automatically Recompiles VDOC Search Tables During Upgrade	27
7.2.10 Ensemble Custom Schema Validation Reports Errors at Compile Time	27
7.2.11 Port Declaration Property for TCP adapter Changed	27
7.2.12 Change in Handling HL7 Message Header Name Field	27
7.2.13 Changes in HL7 Schema Representation	28
8 Ensemble 2013.1	29
8.1 New Features in Ensemble 2013.1	29
8.2 Compatibility Issues for Upgrades to Ensemble 2013.1	29
8.2.1 Simplified X12 Schema Representation	29
8.2.2 Need to Perform Cleanup after Upgrading from Previous Versions	30
8.2.3 Improve Access to ASTM Documents by Building Index	30
8.2.4 Enable Access to DeepSee Dashboard Pages	31
9 Ensemble 2012.2	33
9.1 New Features in Ensemble 2012.2	33
9.2 Compatibility Issues for Upgrades to Ensemble 2012.2	33

9.2.1	New Compiler Behavior for DTL Classes	33
9.2.2	New DTL Classes Created with IGNOREMISSINGSOURCE Parameter Set to True ..	34
9.2.3	Rules Conversion and Upgrade	34
9.2.4	Change in Operator Precedence in Business Rules and Routing Rule Conditions	34
10	Ensemble 2012.1	35
10.1	New Features in Ensemble 2012.1	35
10.2	Compatibility Issues for Upgrades to Ensemble 2012.1	35
10.2.1	New Management Portal User Interface	36
10.2.2	Business Rule Conversion	36
10.2.3	Changes in Rule Log Structure	36
10.2.4	New Dashboard Development Tool	37
10.2.5	New Security Model for Management Portal	37
10.2.6	Changes to Workflow User Interface	37
10.2.7	Updated Selectivity and Extent Size of the Message Warehouse	38
10.2.8	Updated Saved Message Searches	38
10.2.9	Removed CSPX Files from Distribution	39
10.2.10	Updated Search Table Validation	39
10.2.11	New DTL Classes Created with REPORTERRORS Parameter Set to True	39
10.2.12	Updated Legal Character Checking in Configuration Names	40
10.2.13	Change in Inactivity Timeout Behavior	40
10.2.14	Removed Host Monitor from User Interface	40
10.2.15	Improved Notification for Stopping a Running Production	40
10.2.16	Change in \$\$\$EnsSystemError Behavior	41
10.3	Finding Ensemble Options in the New Management Portal	41
11	Ensemble 2010.2	43
11.1	New Features in Ensemble 2010.2	43
11.2	Compatibility Issues for Upgrades to Ensemble 2010.2	43
11.2.1	Remove Support for HL7v2 Framing with XML Text	44
11.2.2	Update Error Handling on HTTP Outbound Adapter	44
11.2.3	Update Error Processing in File Outbound Adapter	44
11.2.4	Change Return Status on HTTP Inbound Adapter	44
11.2.5	Add Requirement to Subclass Message Bank Production	45
11.2.6	Update Disable Behavior of Business Processes	45
12	Ensemble 2010.1	47
12.1	New Features in Ensemble 2010.1	47
12.2	Compatibility Issues for Upgrades to Ensemble 2010.1	47
12.2.1	Relocate RemoveItem() Configuration Method	47
12.2.2	Add Configuration Setting on TCP Counted Outbound Adapter	48
12.2.3	Correct Type Node in HL7 Sequence Manager Global	48
12.2.4	Correct Behavior of HL7 Configuration Framing Setting	48
12.2.5	Add Support for Legacy FTPS Protocol to FTP Adapters	48
12.2.6	Changes in Mapping of Custom Schemas	49
13	Ensemble 2009.1	51
13.1	New Features in Ensemble 2009.1	51
13.2	Compatibility Issues for Upgrades to Ensemble 2009.1	51
13.2.1	Changes in HL7 Storage Structure	52
13.2.2	New ReplyCodeActions Property in Process and Operation Classes	52
13.2.3	New Mechanism for Editing Messages Replaces the %DrawEditForm() Method	53
13.2.4	Increased Alert Level for Data Transformation Errors	53

13.2.5 Changes to Pool Size Configuration Behavior on TCP Service	53
13.2.6 Renamed Column in Statistics Queries	53
13.2.7 Alert Support for Services Invoked Outside Ensemble	53
13.2.8 Changes in Empty Schema Category Behavior	54
14 Ensemble 2008.2	55
14.1 New Features in Ensemble 2008.2	55
14.2 Compatibility Issues for Upgrades to Ensemble 2008.2	56
15 Ensemble 2008.1	57
15.1 New Features in Ensemble 2008.1	57
15.2 Compatibility Issues for Upgrades to Ensemble 2008.1	58
15.2.1 DTL Validation Errors	58
15.2.2 AllowSessions Setting Removed from EnsLib.SOAP.Service	58
16 Older Releases	59
16.1 Ensemble 2007.1	59
16.2 Ensemble 4.0	60
16.3 Ensemble 3.1	61
16.4 Ensemble 3.0	61
16.5 Ensemble 2.1	62
16.6 Ensemble 2.0	62
16.7 Ensemble 1.0	62

About This Book

This book collects information on versions of Ensemble before the current release. This information includes both the new features in those versions and information needed for upgrading to those versions. This book contains the following sections:

- [Ensemble 2017.2](#)
- [Ensemble 2017.1](#)
- [Ensemble 2016.2](#)
- [Ensemble 2016.1](#)
- [Ensemble 2015.2](#)
- [Ensemble 2015.1](#)
- [Ensemble 2014.1](#)
- [Ensemble 2013.1](#)
- [Ensemble 2012.2](#)
- [Ensemble 2012.1](#)
- [Ensemble 2010.2](#)
- [Ensemble 2010.1](#)
- [Ensemble 2009.1](#)
- [Ensemble 2008.2](#)
- [Ensemble 2008.1](#)
- [Older Releases](#)

There is also a detailed [table of contents](#). For the *current* release, see the [Ensemble Release Notes](#).

For a release history of new features and upgrade compatibility issues in Caché, see the following documents:

- [Caché Release Notes and Upgrade Checklist](#)
- [Caché Release Note and Upgrade Checklist Archive](#)

For general information, see [Using InterSystems Documentation](#).

1

Ensemble 2017.2

1.1 New Features in Ensemble 2017.2

Ensemble provides the following new features and enhancements

- [Improved XML Performance](#)
- [DICOM Enhancements](#)
- [XSLT2 Activity in BPL](#)
- [Ensemble is now compatible with Disabling Caché Global Kill](#)
- [Minor Enhancements](#)
- [Caché 2017.2 Features](#)

1.1.1 Improved XML Performance

If you are creating XML Virtual Documents where most of the contents of the XML document are included in the virtual document, you can often obtain substantial performance improvements by converting the XML Virtual Document to an XML-enabled persistent object. This release includes the `EnsLib.XML.Object` package, which provides utility routines to convert an XML document to an XML-enabled persistent object and to convert this object back to an XML document. For details, see “[XML-Enabled Objects Compared to XML Virtual Documents](#)” in the *Ensemble XML Virtual Document Development Guide*.

1.1.2 DICOM Enhancements

In this release Ensemble has the following DICOM enhancements:

- New `AsyncRouter` DICOM sample production. This production demonstrates how to asynchronously route DICOM documents through Ensemble. This production provides a template to build custom productions that asynchronously route DICOM messages. The production is defined in the class `Demo.DICOM.Production.AsyncRouter` in the `ENSDEMO` namespace.
- The DICOM dictionary now conforms with the DICOM Standard 2016E. No existing tags were removed. The changes include new tags, tags with new enumerated values, and two new Value Representations, UC (Unlimited Characters) and OL (Other Long).

1.1.3 XSLT2 Activity in BPL

In this release, you can include an XSLT2 activity in BPL. Set the activity attribute to 2.0 to allow use of XSLT2. You must configure the XSLT2 Gateway in the Ensemble Management Portal.

1.1.4 Ensemble is now compatible with Disabling Caché Global Kill

In some environments, you need to disable Caché top-level global kills for security reasons. In previous releases, this change was not compatible with installing and running Ensemble. In this release, you can install and run Ensemble with top-level Caché global kill disabled.

1.1.5 Minor Enhancements

This release also includes the following minor enhancements:

- When you deploy a production, Ensemble automatically includes generated classes for Record Map and Complex Record Maps. In previous releases, you had to manually add these classes.
- Delimited Record Maps have a new option to output records where all fields are quoted with the quoting character whether or not they contain separator characters.
- The HTTP outbound adapter now has the capability to use a tunnel to connect through a proxy. In previous releases, this feature was available in the SOAP outbound adapter, but not the HTTP outbound adapter.
- The HL7 HTTP business service can now use the CSP port or a special port. In previous releases, it could only use a special port.

1.1.6 Caché 2017.2 Features

Ensemble 2017.2 runs on top of Caché 2017.2. This means that, in addition to changes in Ensemble between 2017.1 and 2017.2, the new Ensemble release includes a large number of updates in the underlying Caché technologies.

Caché 2017.2 has the following important new features described in the *Caché Release Notes and Upgrade Checklist*:

- Parallel dejournaling for mirroring and journal restore
- DeepSee Folder Manager and improvements to dashboard filters
- iKnow cooccurrence search improvements and iKnow performance improvements
- SQL improvements: query auditing, POSIX timestamp, query optimization, optional ANSI SQL operator precedence, and frozen plan evolution

To learn about these and other Caché changes that may relate to your Ensemble environment, see “New and Enhanced Features for Caché 2017.2” in the *Caché Release Notes and Upgrade Checklist*.

1.2 Compatibility Issues for Upgrades to Ensemble 2017.2

Before upgrading Ensemble, first review the product changes in this release that could affect the operation of your existing system. The following sections list the compatibility issues for this and previous releases of Ensemble. In addition to the issues in this release, be sure to also review the issues for each intervening release since you last installed Ensemble:

The following changes in this release may affect the operation of your existing system. Review these following issues before upgrading a previous instance of Ensemble.

- [Do not use JMS Gateway as Message Listener](#)
- [Custom RecordMap Batch Classes may need to be Modified to get Fix](#)
- [BPL xslt Activity Provides More Information in Error Messages](#)
- [Business Service Time to Wait Changes](#)
- [Ensemble Include File Includes Definitions from %sySystem](#)
- [Changes may be Required in PassthroughService Subclass](#)

1.2.1 Do not use JMS Gateway as Message Listener

In this release the JMSGateway class has been deleted from the Java Gateway. This class was intended as an example on how to use the JMS listener in Ensemble. If you have used this class to implement a JMS listener, you should replace this with the mechanism demonstrated in the `EnsLib.JavaGateway.JMSTest` class.

1.2.2 Custom RecordMap Batch Classes may need to be Modified to get Fix

This release fixes a problem where Ensemble would not remove temporary files under certain failure conditions. If you have a custom subclass of `EnsLib.RecordMap.Batcher` or `EnsLib.RecordMap.Operation.BatchStandard` and have implemented the `OnFailureTimeout()` method, you can add a call to `##super()` in that method to get this fix.

1.2.3 BPL xslt Activity Provides More Information in Error Messages

The BPL activity can now handle XSLT 2.0 transformations. If using XSLT 1.0 there may be more information in any error messages.

1.2.4 Business Service Time to Wait Changes

Ensemble Business Services calculation of time to wait for next call interval now discounts any fractional part that is 1/10th second or less. This has been done since a positive wait for next call interval time is rounded up to the next second. For example 0.000001 will not lead to a wait of 1 second.

1.2.5 Ensemble Include File Includes Definitions from %sySystem

The include file `Ensemble.inc` now includes `%sySystem`. If you define any macros custom code, you should check for macro name collisions.

1.2.6 Changes may be Required in PassthroughService Subclass

If you have a custom subclass of `EnsLib.File.PassthroughService` with an override of `CONTAINERCLASS` or with an override of the adapter you may need to override the new `removeStreamFromInput()` method within the subclass.

2

Ensemble 2017.1

2.1 New Features in Ensemble 2017.1

Ensemble provides minor feature and efficiency improvements. These include the following:

- Improved Efficiency for Message Search
- Record Mapper Can Ignore Fields when Loading Files
- Ability to Create Namespaces that are not Enabled for Ensemble
- Minor Enhancements:
 - Message traces include information from rule logs.
 - Activity Monitor includes message class or vDoc type in the site dimension column for some messages.
 - Support for EDIFACT documents with explicit indication of level of repetition.
 - Showing connection lines in Production Configuration page between router and delegate targets.
 - TCP internal event notification improvements.
 - DICOM Duplex TCP adapter supports both incoming and outgoing SSL and allows you to set the local interface and connection queue size.
 - Improvements to processing Super Session IDs.
 - The Search Table Unselective property is now also used to improve the efficiency of searches in the Enterprise Message Bank.

2.2 Compatibility Issues for Upgrades to Ensemble 2017.1

The following changes in this release may affect the operation of your existing system. Review these following issues before upgrading a previous instance of Ensemble.

- [Business Process Lock Changes](#)
- [XML VDoc Handling of Missing Values Changes](#)
- [XML VDoc Child of Repeating Parent Changes](#)

- [Ensemble Java Gateway Defaults to JDK 1.8](#)
- [Ensemble DotNet and Java Gateways Support Validated Connections](#)
- [Search Tables Now Handle Vertical Bars](#)

2.2.1 Business Process Lock Changes

In this release we have improved handling locks in business processes that are coded to allow multiple asynchronous responses. In previous releases, if a part of the business process exceeded the system lock timeout, the asynchronous response could be discarded. In this release, instead of discarding the response, Ensemble queues the response to the business process.

2.2.2 XML VDoc Handling of Missing Values Changes

In previous releases, transformations could insert an empty string value in the target where the property in the source was missing. In this release the value in the target is missing and not the empty string. If your transformation relies on the missing value in the target, you can explicitly set the target property to an empty string.

2.2.3 XML VDoc Child of Repeating Parent Changes

In previous releases, if the XSD schema defined a repeating parent that contained a repeating child and there was only a single child element in the XML document, the VDoc could define the child as a member of the repeating parent. This prevents you from accessing any child element other than the first. In this release, XML VDoc correctly parses the repeating element. If you have written code to explicitly handle the previous structure, you should modify your code to handle the corrected structure.

2.2.4 Ensemble Java Gateway Defaults to JDK 1.8

In previous versions, the Ensemble Java Gateway defaulted to JDK 1.7, but in this release it defaults to JDK 1.8. If you need to continue using JDK 1.7, specify that version in the Java Gateway settings.

2.2.5 Ensemble DotNet and Java Gateways Support Validated Connections

In this release, you can start the Ensemble Java and DotNet gateways and require passphrases. There are also changes to specifying the address for the gateway and starting the gateway for use by another system. See “Java and Gateway Changes” for 2017.1 in the *Caché Recent Upgrade Checklists* for details.

2.2.6 Search Tables Now Handle Vertical Bars

In previous releases, Ensemble did not allow a vertical bar (|) in any of values indexed by a search table, because double and single vertical bars both interfered with Ensemble internals.

In this release, this limitation has been removed, and any vertical bar is replaced by a plus sign when the data is indexed. Recompile any search table classes to enable them to index new data correctly. Note that this has no effect on the existing contents of the search tables. Also, search for a plus sign rather than a vertical bar, when you use the search table. For example, to search for a message that contains my|string, use my+string.

3

Ensemble 2016.2

3.1 New Features in Ensemble 2016.2

Ensemble 2016.2 provides minor feature and efficiency improvements. These include the following:

- Export event logs to a file for reporting and analysis. You can export to a text, delimited, HTML, or XML file.
- Ability to configure ASTM TCP business service and operation to work with servers that do not provide a response.
- Improved loading of HL7 schemas avoiding temporarily undefined schema.
- Improved display of visual trace with many log entries. If there are more than 2,000 log entries, the display includes the first 1,000 and the last 1,000.
- Improved efficiency loading complex XML virtual document schemas.
- XML VDoc Data Transformations configured with **Ignore missing source segments** selected will now ignore any actions that access a missing element. In previous versions, these actions would cause errors. You should recompile the Data Transformation to get the new behavior.
- Better integration with source control to identify read-only definitions.
- Ability to use outbound email adapter to send messages with only Bcc: recipients specified in the Ensemble Portal settings. At least one To: or Cc: address must be specified for the outbound email adapter to send the message, but this can be set in the adapter or in the email message.
- New RecordMapBatch sample in ENSDEMO, which shows how to improve efficiency by loading multiple records from a file in a single step.
- Improved performance of production configuration page in management portal.

3.2 Compatibility Issues for Upgrades to Ensemble 2016.2

The following changes in this release may affect the operation of your existing system. Review these following issues before upgrading a previous instance of Ensemble.

- [Production Status Hover Tip May Be Stale](#)
- [IsEnsembleNamespace Method Returns Information About Specified Namespace](#)
- [Efficiency Improvements Changed VDoc XSD Schema Structure](#)

- [Ensemble Editors Now Respect Source Control Readonly Property](#)
- [Ensemble X12 Business Service Limits 999 Implementation Acknowledgement to Allowed Transaction Set Codes](#)

3.2.1 Production Status Hover Tip May Be Stale

The Production Configuration page checks the status of the production and tests if any component needs updating. These checks temporarily lock the production's runtime data. In this release, we have improved efficiency by minimizing the locks. A consequence of this change is that the hover text indicating that a production needs updating may be out of date and not reflect the current state. If you select the **Update** button, the data is refreshed and the update uses the current data.

3.2.2 IsEnsembleNamespace Method Returns Information About Specified Namespace

In this release, the `##class(%Library.EnsembleMgr).IsEnsembleNamespace()` method takes a parameter and returns whether the specified namespace is enabled for Ensemble. In previous releases, this method always returned the status of the current namespace. If you have code that specifies a parameter, in previous releases, it would have ignored the parameter and returned the status of the current namespace. In this release, it returns the status of the specified namespace. If no parameter is specified, the behavior is unchanged; the method returns the status of the current namespace.

Note: If the user does not have access to the specified namespace, this method always returns 0.

3.2.3 Efficiency Improvements Changed VDoc XSD Schema Structure

This release improves the efficiency of storing XSD by eliminating unneeded data. This can also substantially reduce the amount of data that needs to be journaled. These internal structures should only be accessed by InterSystems internal code. If you have written code that directly accesses these structures, you should modify it to use the public APIs.

Note: If a large schema is re-imported, it is possible that large journal records will be produced as the unnecessary schema data is removed.

3.2.4 Ensemble Editors Now Respect Source Control Readonly Property

Although the Ensemble editors, such as the DTL and BPL editors, supported source control systems, they did not prevent you from editing files that were not checked out in the source control system. Although Ensemble would allow you to edit the file, you would not be able to check in your changes. In this release, if source control is enabled for a definition, Ensemble will only allow you to edit the file if you have checked it out and the source control system provides write access.

To make the document editable, check it out and then refresh the page. If you wish to edit a definition that is not checked out, change the name by doing a save as. In the Record Mapper and Complex Record Mapper, you can accomplish this by modifying the RecordMap or Complex RecordMap name before saving.

The Ensemble editors are:

- BPL Editor
- DTL Editor
- Data Lookup Tables
- Rule Editor
- Record Mapper
- Complex Record Mapper

3.2.5 Ensemble X12 Business Service Limits 999 Implementation Acknowledgement to Allowed Transaction Set Codes

In this release the X12 Business Service generates a 999 implementation acknowledgement only for only for HIPAA_5010 (or later versions) batch transactions that define it as a valid reply. In previous releases, the 999 implementation acknowledgement was returned for other transaction sets. In previous releases, the X12 Business Service also returned a 999 implementation acknowledgement for other HIPAA_5010 transaction sets.

X12 Business Services will now construct 999 responses only for X12 batch transactions which have one of the following as their Industry Identifier Code (associated Transaction Set Identifier Codes indicated in parentheses):

- 005010X187 (269)
- 005010X279 (270/271)
- 005030X209 (274)
- 005040X254 (275)
- 005010X212 (276/277)
- 005010X214 (277CA)
- 005010X217 (278)
- 005010X218 (820)
- 005010X220 (834)
- 005010X221 (835)
- 005010X222 (837P)
- 005010X223 (837I)
- 005010X224 (837D)

If you have code that expects the 999 implementation acknowledgement for other transaction sets, you can either modify this code or override the `OnConstructReply()` method to revert the X12 Business Service to the previous behavior.

4

Ensemble 2016.1

4.1 New Features in Ensemble 2016.1

Ensemble 2016.1 introduced the following new features and enhancements:

- [Service Registry and Enhanced Tools to Use Ensemble as an ESB](#)
- [Activity Volume Statistics and Monitoring](#)
- [SuperSession Identity Propagated Between Productions](#)
- [Incremental X12 Enhancements](#)
- [Minor Improvements to Ensemble Graphic Editors](#)
- [Production and Component Restart Improvements](#)
- [Message Bank Enhancements](#)
- [New Filter on Message Bank Viewer](#)
- [Rule Can Access Information About Transformation](#)

4.2 Compatibility Issues for Upgrades to Ensemble 2016.1

The following changes in this release may affect the operation of your existing system. Review these following issues before upgrading a previous instance of Ensemble.

- [EnsLib.REST.Service Handles Query Parameters Differently](#)
- [Upgrading Ensemble Sets _Ensemble User to Not Expire](#)
- [Ensemble Checks that Default Database Allows Write Access](#)
- [Business Process Termination Error Now Includes Text from Underlying Error](#)
- [Inbound TCP Adapters May Report Additional Errors on Startup](#)
- [Option to Improve Restart with Suspended Messages Changes Storage Location](#)
- [Improved Validation of ebXML Messages Changes Behavior](#)
- [Additional Log Messages when Disconnecting Java Gateway](#)

- [Email Outbound Adapter Defaults Change](#)
- [FTP Failure to Delete Network File Handled Without Reprocessing File](#)
- [Router Sets Headers Correctly After Timeout](#)
- [BPL No Longer Allows Two Calls With the Same Name](#)
- [TCP Counted Adapter Now Checks Incoming Length](#)
- [Disabling a Business Process with a Pool Size of 0](#)
- [Improved Check of Permissions for Purging Alerts](#)
- [XML File and FTP Operations Write Search Table Data](#)
- [Purging Completed Business Process Instances with Null TimeCreated Values](#)
- For details on compatibility issues in upgrading Caché to version 2016.1, see the *Caché Release Note and Upgrade Checklist Archive*.

4.2.1 EnsLib.REST.Service Handles Query Parameters Differently

In this release subclasses of EnsLib.REST.Service need to get query parameters from attributes not from the URL. In previous releases, it was also possible to get it from the URL when the incoming request came through the CSP port. This change was required to fix problems matching the incoming URL with the URLMap. The query parameters are available as attributes of the stream object passed to OnProcessInput or from the %request object.

Note that we recommend directly subclassing %CSP.REST and not subclassing EnsLib.REST.Service. If you directly subclass %CSP.REST you can use the full features of that class. In order to use a subclass of %CSP.REST in an Ensemble production you need to call CreateBusinessService.

4.2.2 Upgrading Ensemble Sets _Ensemble User to Not Expire

In most cases, the _Ensemble user, which is used by Ensemble internally should be set to not expire and to have the password not expire. In this and future releases, upgrading Ensemble will reset the AccountNeverExpires and PasswordNeverExpires properties to true and notes the changes in the ensinstall log. If your environment requires that one or both of these properties be set to false, you must reset them after upgrading Ensemble.

4.2.3 Business Process Termination Error Now Includes Text from Underlying Error

In this release when a business process encounters an error, the termination error includes the text from the underlying error. If you have error handling code that is comparing the error text, you may have to modify your code.

4.2.4 Ensemble Checks that Default Database Allows Write Access

In this release, Ensemble checks the access to the default database before starting the production. If the default database is read-only, the management portal displays an error and does not start the production. In most cases, this condition is caused by a configuration error and the production would have failed after startup. Although it is not a recommended practice, it was possible to use global mapping to create a production that would work with a read-only default database. This configuration will not work in this release. You must modify the configuration so that the default database allows write access before starting the production.

4.2.5 Inbound TCP Adapters May Report Additional Errors on Startup

In previous releases the inbound and outbound TCP adapters set to be always connected (`StayConnected = -1`) handled startup differently. If an initial connection is not made within the alert grace Period, the outbound adapter reported an error but the inbound adapter did not. The inbound adapter only reported an error if the adapter was set to have only one listener and an existing connection ended.

In this release the TCP Inbound Adapter set to be always connected behaves the same during start up as the outbound adapter. It reports an error if the initial connection is not made within the alert grace period. This error may be triggered when systems are being brought back online after an outage leading to an Alert On Error if configured. If there is a lag in making the TCP connection in the normal startup, you can avoid this error by increasing the alert grace period.

4.2.6 Option to Improve Restart with Suspended Messages Changes Storage Location

In previous releases, when a production was stopped, any asynchronous messages on the `^Ens.Queue` global queue was moved to the `^Ens.Suspended` queue and when the production was restarted, they were moved back. For productions with many messages in the queue, this behavior could cause stopping and restarting a production to be slow. In this release, you have the option to avoid this move and the consequent delay in restarting. To avoid the move of suspended messages, set `^Ens.Configuration("Queues","KeepInQueues")=1` per namespace. The default is 0 which is to leave the current behavior.

This change does not have a compatibility impact on your code in your production even if you request the new behavior. But, if you have code that examines suspended messages stored in the database globals while the production is stopped, you should update your code to handle the new option.

4.2.7 Improved Validation of ebXML Messages Changes Behavior

This release includes improved validation of ebXML messages; consequently, some messages that were not valid passed validation in previous releases, but are now treated as errors. Specifically, an ebXML message is considered not valid if both the `MessageId` and `ConversationId` are empty. In previous releases, these messages could pass validation.

4.2.8 Additional Log Messages when Disconnecting Java Gateway

In order to successfully close the Java Gateway after a ping operation, this release sends additional send and receive messages to the gateway. This causes additional entries in the log.

4.2.9 Email Outbound Adapter Defaults Change

In previous releases, if there were any errors in the distribution list, the message would not be sent to any recipients. In this release, the message is sent to all recipients except to those addresses with an error. To have the outbound adapter return to the previous behavior and not send email to any recipient, set the `ContinueAfterBadSend` setting to false.

4.2.10 FTP Failure to Delete Network File Handled Without Reprocessing File

If an FTP adapter is configured to delete the file on the server and fails to then Ensemble should consider the file processed and not reprocess it. In past releases, this behavior was correct if the adapter could not delete the file because of permissions. But, if the adapter could not delete the file because of network problems, it incorrectly treated the file as new and reprocessed it. This problem has been fixed in this release. The adapter will not reprocess the file. If you have coded your production to deal with the past incorrect behavior, you should remove this code.

4.2.11 Router Sets Message Headers Correctly After Timeout

In previous versions, if a router timed out before processing a message, it would not set the message header correctly and the message header could not be purged. This error is fixed in this release. If you have developed a procedure to handle the unpurged messages, it is no longer needed for this condition.

4.2.12 BPL No Longer Allows Two Calls With the Same Name

In previous versions, the BPL wizard did not detect the error of having two calls with the same name. This condition could cause an unexpected call. In this version for synchronous calls that have a response, this error is detected by the BPL wizard. Consequently, BPLs that had this error would have compiled successfully in previous versions, but will trigger an error in this version.

4.2.13 TCP Counted Adapter Now Checks Incoming Length

In previous releases, the TCP counted adapter ignored the incoming length value if there was no character encoding; consequently, if the actual length did not match the specified length, the adapter accepted the message. In this release, the adapter makes this check and rejects the incorrectly formatted message. This change makes the behavior of the TCP counted adapter with no character encoding match its behavior with character encoding, where it has always checked the incoming length.

4.2.14 Disabling a Business Process with a Pool Size of 0

In previous releases if a business process with a pool size of 0 was disabled, all business processes that use the shared actor pool would be disabled. Typically, you only want to disable a specific business process, not all that use the shared actor pool. In this release if you attempt to disable a business process with a pool size of 0, Ensemble displays a warning and the business process is not disabled.

Note: If you attempt to disable a business process with a pool size of 0, the business process is not disabled and is displayed as green on the production configuration diagram, but the enabled check box is not selected.

If you do want to disable a business process with a pool size of 0, you can do this by setting the pool size to a positive integer. If you do want to stop all business processes using the shared actor pool, set the **Actor Pool Size** to 0 in the production settings tab.

If you attempt to create a new business process that is disabled and that has a pool size of 0, the business process is created disabled, but the pool size is set to 1. This situation occurs if you:

- Add a new business process with a pool size of 0 and disabled.
- Copy an existing business process with a pool size of 0.

In these cases, the disabled business process is set to have a pool size of 1 instead of the specified 0.

4.2.15 Improved Check of Permissions for Purging Alerts

The Ensemble Event Log page was failing to enforce the security requirement that the user needs USE permissions on the %Ens_Purge resource to purge event log entries. This is now corrected and the security requirement is now enforced. If a user without the permission clicks on Purge the alert "You are not permitted to perform this action" will be shown and the purge will not be carried out.

4.2.16 XML File and FTP Operations Write Search Table Data

In previous releases, if you specified a search table in an `EnsLib.EDI.XML.Operation.FileOperation` or `EnsLib.EDI.XML.Operation.FTPOperation`, the search table index was not created. In this release, the search table index is created. Note that when the message bodies are purged, the search table data is also purged.

4.2.17 Purging Completed Business Process Instances with Null TimeCreated Values

In previous releases completed business process instances might not be purged if the `TimeCreated` value was null. This situation only occurred if the debug flag was set to retain business process instances and custom code updated the `TimeCreated` value, which should not be set by user code. In this release, even in this case, the completed business process will be purged as expected.

5

Ensemble 2015.2

5.1 New Features in Ensemble 2015.2

Ensemble 2015.2 introduced the following minor new features and enhancements:

- Improved Efficiency with Pass-Through Generic InProc Operations
- Message Bank Improvements
- Production Monitor Shows Retry Status
- Purge Improvements
- Transformation and Subtransformation Restrictions Removed

5.2 Compatibility Issues for Upgrades to Ensemble 2015.2

The following changes in this release may affect the operation of your existing system. Review these following issues before upgrading a previous instance of Ensemble.

- [Upgrading Enterprise Message Bank Requires Rebuilding Indices](#)

5.2.1 Upgrading Enterprise Message Bank Requires Rebuilding Indices

If you are upgrading an Ensemble Message Bank system from Ensemble 2015.1 or earlier to Ensemble 2015.2 or later, you must disable and rebuild four new indices on the Message Bank System that are created during the upgrade. Until you disable the indices, Message Bank search will not find any messages that were sent to the Message Bank before the upgrade. Once you have rebuilt the indices, the Message Bank will use them and you will get improved performance on message search.

Ensemble 2015.2 includes a development change that improved Message Bank searches by replacing date-based searches with the equivalent searches based on IDs. This substantially improves the efficiency of searches especially when searching on date ranges. This change is dependent on new indices that are created when you upgrade Ensemble, but, by default, the new indices are only populated for messages received after the upgrade. This means that Message Bank search will only find these new messages and will not find old messages until you rebuild the indices.

Rebuilding indices is covered in the Caché documentation in “Building Indices on a READ and WRITE Active System” in *Caché SQL Optimization Guide*. The procedure described in this compatibility issue is slightly different from the procedures described there.

This procedure assumes that your Message Bank production is live and is actively receiving messages during this procedure. You only have to perform this procedure on the Message Bank system and do not have to make any changes to the client systems that are sending messages to the Message Bank.

To disable the four indices that were created during the upgrade, rebuild the indices, and re-enable them, follow this procedure:

1. Inactivate the indices with the following:

```
Do $SYSTEM.SQL.SetMapSelectability("Ens_Enterprise_MsgBank.MessageHeader", "NodeIdTime", 0)
Do $SYSTEM.SQL.SetMapSelectability("Ens_Enterprise_MsgBank.MessageHeader", "NodeId", 0)
Do $SYSTEM.SQL.SetMapSelectability("Ens_Enterprise_MsgBank.MessageHeader", "NTrg", 0)
Do $SYSTEM.SQL.SetMapSelectability("Ens_Enterprise_MsgBank.MessageHeader", "NSrc", 0)
```

2. If new messages have been added to the message bank since the upgrade, you should purge any cached queries. See “Purging Cached Queries ” in *Caché SQL Optimization Guide*. At this point, queries should work using the other indices on the Message Bank. They will not have any improved efficiency, but they will return old as well as new messages.
3. You can skip steps 2 through 5 as listed In “Building Indices on a READ and WRITE Active System”. The Message Bank code performs these steps automatically.
4. If you are upgrading to Ensemble 2016.2 or later, use the %BuildIndices method to rebuild the indices.

```
Set tSC =
##class(Ens.Enterprise.MsgBank.MessageHeader).%BuildIndices($LB("NodeIdTime", "NSrc", "NTrg", "NodeId"), ...)
```

See the %Library.Persistent %**BuildIndices()** method for a description of the method parameters. Since this method can take a long time to complete, you may want to build the indices in batches using the pStartID and pEndID parameters.

5. If you are upgrading to Ensemble 2015.2 or Ensemble 2016.1, use the %ConstructIndicesParallel() method to build the index or indices. This method can take a long time to complete depending on the size of the Message Bank. [This is step 6 in the original procedure.]

```
Set tSC=##class(Ens.Enterprise.MsgBank.MessageHeader).%ConstructIndicesParallel(, , 0, 0, 2, 0)
```

These parameters mean:

- pSortBegin=0 do not use the SortBegin feature (using SortBegin is incompatible with using row level locking).
- pDroneCount=0 allows the method to set the number of background jobs.
- pLockFlag=2 says to use row level locking.
- pJournalFlag=0 means the index building will not be journaled. If the system fails during the index building, you will have to restart by re-entering this call.

This call will only build the four new indices because the %ConstructIndicesParallel uses the INDEXBUILDERFILTER class parameter, which specifies these four indices.

6. Enable the index to be used in search with the following commands:

```
Do $SYSTEM.SQL.SetMapSelectability("Ens_Enterprise_MsgBank.MessageHeader", "NodeIdTime", 1)
Do $SYSTEM.SQL.SetMapSelectability("Ens_Enterprise_MsgBank.MessageHeader", "NodeId", 1)
Do $SYSTEM.SQL.SetMapSelectability("Ens_Enterprise_MsgBank.MessageHeader", "NTrg", 1)
Do $SYSTEM.SQL.SetMapSelectability("Ens_Enterprise_MsgBank.MessageHeader", "NSrc", 1)
```

6

Ensemble 2015.1

6.1 New Features in Ensemble 2015.1

Ensemble 2015.1 and introduced the following new features and enhancements:

- Push notifications for mobile devices
- Improved database storage for namespaces
- REST service enhancement
- Pass-through calls can use the CSP gateway
- Stateless calls to Java
- Enterprise and production monitor enhancements

6.2 Compatibility Issues for Upgrades to Ensemble 2015.1

The following changes in this release may affect the operation of your existing system. Review these following issues before upgrading a previous instance of Ensemble.

- [New Global and Database Used to Store Credentials Passwords](#)
- [New Databases and Changes to Where Temporary Globals are Stored](#)
- [JDBC SQL Returns Large Objects as Streams instead of Strings](#)
- [Lookup Table Import and Export Format Changed to be Compatible with Studio](#)
- [Deleting X12 Objects Deletes Child Objects](#)
- [DTL foreach Error Reported Correctly](#)

6.2.1 New Global and Database Used to Store Credentials Passwords

When you create a new namespace, credentials passwords are stored in a secondary database, which is, by default, not accessible to user accounts. For most productions this does not create a compatibility issue because Ensemble provides access to the password when it is needed by the business host. This change provides increased protection for stored passwords. Note that InterSystems recommends that you encrypt any database containing credentials passwords.

There are some conditions where this change can cause a compatibility issue. This section describes these conditions and ways you can avoid problems.

Note: Some Ensemble licenses limit the number of databases allowed. This new credentials database is counted towards the limit.

When you create a new namespace, Ensemble creates a secondary database and names it by appending `SECONDARY` to the globals database name. For example, if you create the `INVENT` namespace with the `INVENTG` database for globals and the `INVENTR` database for routines, Ensemble also creates the `INVENTGSECONDARY` database for credentials passwords. In previous releases, Ensemble stored the credentials password in the `^Ens.Conf.CredentialsD`, but in this release, Ensemble stores credentials passwords in the global `^Ens.SecondaryData.Password` and maps that global to the secondary database.

For existing namespaces, upgrading Ensemble does not create the new secondary database, but when you first access credentials, the passwords are moved to the new global. If you call the `%Library.EnsembleMgr:CreateNewDBForSecondary()` method, Ensemble creates the secondary database for credentials passwords, migrates the password to the new secondary database, and maps the `^Ens.SecondaryData.Password` to the secondary database. You should only call the `CreateNewDBForSecondary()` method when the production has been stopped.

In most cases, these changes do not impact production code or accessing credentials through the management portal, but it does create the following compatibility issues:

- Any job running code to read credentials now needs resources protecting the secondary password database. For example, the `INVENTGSECONDARY` is protected by the resource `%DB_INVENTGSECONDARY`, to read the credentials a process must have read access to the `%DB_INVENTGSECONDARY` resource. Note that jobs started as part of the production run under the user `_ENSEMBLE` and these jobs are able to read credentials
- Any job running code to write credentials now needs `%Ens_Credentials:"W"` resource. Note that the management portal escalates permissions when writing credentials. Consequently, you do not have to change the permissions of portal users who are able to enter credentials with the previous version of Ensemble.
- If a web application reads or writes credentials, including by calling an `InProc` business operation, the web application needs to be given the necessary resources, at least for the duration of the read or write of the credentials.
- If code is running in another context and needs access to the credentials password, the user account needs to have the necessary resources.
- If you call the `CreateBusinessService()` of `Ens.Director` to create a business service that sends a message directly to an `InProc` operation that requires access to the credentials password, the user account needs to have the necessary resources.
- If in an existing namespace you have created a mapping that moves `^Ens.Conf.CredentialsD` to another database, you must create the same mapping for `^Ens.SecondaryData.Password`.

Note: If you are running an instance of HealthShare, it does not create the secondary database for credentials passwords unless you explicitly call the `CreateNewDBForSecondary()` method. Do not call this method for a namespace used in a HealthShare Information Exchange.

6.2.2 New Databases and Changes to Where Temporary Globals are Stored

The `Runtime` and `JobStatus` data is now stored in the global `^CacheTemp.EnsRuntimeAppData` subscribed by namespace. In previous versions, this data was stored in the global `^Ens.RuntimeAppData`. In addition, metrics data, which is used for display in the production monitor, is now stored in the global `^CacheTemp.EnsMetrics`. In previous versions, this data was stored in the global `^Ens.Metrics`. In most cases, these changes do not cause a compatibility issue, but you should ensure the following:

- No productions are in the troubled state when you upgrade to a version of Ensemble that has this change.

- If your code explicitly accesses these globals, you must update the code to access the data in the new location.

Since these globals are now stored in a non-journaled database, the values are not available to mirror members.

Note: Some Ensemble licenses limit the number of databases allowed. This new database for temporary globals is counted towards the limit.

6.2.3 JDBC SQL Returns Large Objects as Streams instead of Strings

When using JDBC SQL adapters, stored Procedure large object (LOB) output parameters are now returned as streams. In previous versions, these output parameters were returned as strings, which caused errors when the LOB exceeded the maximum string size. You should modify any code that accesses these output parameters to handle streams instead of strings.

As a temporary workaround, you can set globals to return string values for these output parameters. But even with these globals set, if the size of the LOB exceeds the maximum string size, the output parameters are returned as streams.

To configure Ensemble to return LOB output parameters as strings when size permits, set one of the following globals to 1:

- `^Ens.Config("JDBC", "LOBasString", ConfigName)`
where *ConfigName* is the name of the business service, process, or operation.
- `^Ens.Config("JDBC", "LOBasString")`

6.2.4 Lookup Table Import and Export Format Changed to be Compatible with Studio

The Ensemble portal **Import** and **Export** buttons now use the same file format as Studio. In previous versions of Ensemble the portal export format was incompatible with the Studio export format.

If you exported a lookup table with the portal from a previous version of Ensemble, you must use the **Import Legacy** button on Ensemble 2015.1 or later to import it. If you are exporting a lookup table from Ensemble 2015.1 or later and intend to import it with an earlier version of Ensemble, you must import it on the earlier version using Studio; you cannot use the portal import.

6.2.5 Deleting X12 Objects Deletes Child Objects

In previous releases, deleting an X12 object would not delete its child objects. In this release, when you delete an X12 object, all its child objects are deleted.

6.2.6 DTL foreach Error Reported Correctly

Under some circumstances in previous releases, a Data Transformation would not report an error if you included the iterator key within the parentheses in a foreach. This error is now correctly reported.

7

Ensemble 2014.1

7.1 New Features in Ensemble 2014.1

Ensemble 2014.1 introduced the following new features and enhancements:

- REST services and operations
- Pass-through HTTP, REST, and SOAP services and operations
- Alert management
- HL7 Version 2 field-level validation and schema editor enhancements
- Enhanced production export and import from the portal
- XML VDoc enhancements

7.2 Compatibility Issues for Upgrades to Ensemble 2014.1

The following changes in this release may affect the operation of your existing system. Review these following issues before upgrading a previous instance of Ensemble.

- [HL7 Sequence Manager Automatically Adds Index](#)
- [New Security Resource to Control Exporting Deployment Packages](#)
- [Ensemble HL7 Supports Long Strings](#)
- [GetProductionStatus Returns Improved Status Information](#)
- [Ens.Settings has New GetSettingRow Method](#)
- [Ensemble CSP Pages Set UseCookies Property](#)
- [Viewing EDI Raw Contents Inserts Newlines for Consistency](#)
- [Improved Handling of Spaces in FTP File Names](#)
- [Ensemble Automatically Recompiles VDOC Search Tables During Upgrade](#)
- [Ensemble Custom Schema Validation Reports Errors at Compile Time](#)
- [Port Declaration Property for TCP adapter Changed](#)

- [Change in Handling HL7 Message Header Name Schema Field](#)
- [Changes in HL7 Schema Representation](#)

Also review the *Caché Release Note and Upgrade Checklist Archive*.

7.2.1 HL7 Sequence Manager Automatically Adds Index

The HL7 Sequence Manager, which ensures that HL7 messages are processed in the correct sequence, has added an index to improve efficiency. If your production has a business process that has a class of `EnsLib.HL7.SequenceManager` or a subclass of it, it automatically creates an index to improve performance when it processes its first message after the upgrade to Ensemble 2014.1.

You can manually create this index after an upgrade to Ensemble 2014.1 before this business process starts processing messages by doing the following:

1. Ensuring that the Sequence Manager business process is stopped.
2. Start the Ensemble terminal and set it to the namespace used for the production.
3. Enter the following command:

```
Set tSC=##class(EnsLib.HL7.SM.Version).Upgrade()
```

4. Check tSC for an error status to ensure that the command executed correctly.

7.2.2 New Security Resource to Control Exporting Deployment Packages

In this release, the `%Ens_DeploymentPkg` security resource controls the ability to export a deployment package and the `%Ens_Deploy` security resource controls the ability to apply a deployment package to a production. In previous releases `%Ens_Deploy` controlled both of these abilities.

7.2.3 Ensemble HL7 Supports Long Strings

In this release, the Ensemble HL7 framework can take advantage of long strings when they are enabled for a system. See “Support for Long String Operations” in *Caché Programming Orientation Guide* for details on enabling long strings. To maintain the existing 32K limit on segment storage even if long strings are enabled, you can set `^Ens.Config("HL7-NoLongStrings") = 1`.

7.2.4 GetProductionStatus Returns Improved Status Information

In this release, the `GetProductionStatus()` method may return localized text providing status information. If you are testing the return value you should use the `StatusEnum` field, which contains the numeric state value and will be consistent across all locales.

7.2.5 Ens.Settings has New GetSettingRow Method

The `Ens.Settings` class has a new `GetSettingRow()` generated method. If you have extended the `Ens.Settings` class and have previously defined a method with this name in the class, you should update your code. Rename your extension method before recompiling your code to avoid the method name collision.

7.2.6 Ensemble CSP Pages Set UseCookies Property

Ensemble now sets the `UseCookies` property to “Always” when the system is upgraded or a new CSP page is created.

7.2.7 Viewing EDI Raw Contents Inserts Newlines for Consistency

If you display the raw contents of an EDI message, Ensemble ensures that there is always a newline character between segments. In previous versions, some but not all segments were separated by newline characters. Consequently, if you copy and paste the text in the raw content display, the data from this release may contain newline characters that were not present in the data produced by the previous release.

7.2.8 Improved Handling of Spaces in FTP File Names

The FTP Inbound Adapter assumes that any spaces after the FTP file name are part of the file name and do not delimit other fields. This change allows the FTP Inbound Adapter to handle FTP servers that may include spaces in file names. The FTP Inbound Adapter, consequently, cannot support an FTP server that uses a space to terminate a file name and includes other fields after the file name.

7.2.9 Ensemble Automatically Recompiles VDOC Search Tables During Upgrade

When you upgrade to Ensemble 2014.1, it automatically recompiles all user-defined subclasses of `Ens.VDoc.SearchTable` in all Ensemble namespaces.

7.2.10 Ensemble Custom Schema Validation Reports Errors at Compile Time

In 2014.1, the validation of HL7 custom schemas has been improved and more errors will be shown when loading or saving a schema. For example, a message structure that was missing a closing `}` would show no errors when saved under 2013.1 but will give an error “ERROR #5002: Cache error: Mismatched braces '{' and '}' in message structure ...” when loaded or saved on 2014.1. If a database containing an invalid message structure is upgraded, there will be no error during upgrade, but any later attempt to save the schema from studio or the schema editor will give an error. The run time behavior of a schema with this type of error is unpredictable and the newly identified error should be corrected. You should consider correcting the error on any prior version that is still running with the schema in question.

7.2.11 Port Declaration Property for TCP adapter Changed

In 2014.1, Port property used by `EnsLib.TCP.InboundAdapter` is declared as type `Ens.DataType.TCPAgentPort`, which has its own declaration and extends `%String`. In previous releases, the Port property was declared in the superclass `EnsLib.TCP.Common` as type `%Integer`. If you have developed a custom adapter class that subclasses `EnsLib.TCP.InboundAdapter` and declares Port as `%Integer` with `MAXVAL` and `MINVAL` qualifiers, compiling the class produces an error. To fix this error, you can declare Port as `Ens.DataType.TCPAgentPort` and remove any `MAXVAL` and `MINVAL` qualifiers.

7.2.12 Change in Handling HL7 Message Header Name Field

Ensemble calculates the HL7 message name based on the value of the MSH:9 field. This change modifies how Ensemble treats the MSH:9.3 subfield.

HL7 messages may use MSH:9.3 subfield in one of two ways: 1) to qualify the message name, typically with a number, or 2) to specify the message structure type. For example, if MSH:9 has the value “ORM^001^5”, MSH:9.3 has the value “5”, which describes a subtype of the ORM_001 message. But if MSH:9 has the value “ADT^A08^ADT_A01”, MSH:9.3 has the value “ADT_A01”, which specifies that the ADT_A08 message has the ADT_A01 structure type.

When Ensemble processes the MSH:9 field it tests whether the MSH:9.3 subfield has a simple value, such as “5”, or a structured value, such as “ADT_A01”. If it has a simple value, Ensemble appends the value to the message name and uses the message name to find the structure in the schema and to set the value of the Name property. If MSH:9.3 has a complex

value, Ensemble ignores the value. Since Ensemble uses the schema definition to reliably get the message structure type, it does not need to use this value.

If you set the global `^Ens.Config("HL7NamePropOld")=1`, Ensemble uses its original logic in handling MSH:9 and always appends the MSH:9.3 value to the message name, even if it represents a structure type. Set this global in the namespace used for the HL7 production.

7.2.13 Changes in HL7 Schema Representation

Ensemble 2014.1 includes the following enhancements in the HL7 schema definitions:

- **Segment length**—In previous versions, you could only define the length of segments, which set the maximum length allowed. In this release, you can specify the minimum and maximum length of segments.
- **Components of data structure**—In this release, you can specify the component minimum length and maximum length and whether the component is optional or required.

To implement these changes, we made changes in how the HL7 schema elements are stored internally in globals and minor changes to their XML representation. In most cases, these changes will not cause compatibility problems. However, if you are doing any of the following, you should check for compatibility issues:

- Accessing the `^EnsHL7.Schema` or `^EnsHL7.Description` globals directly. These are internal Ensemble structures and may be changed to implement new Ensemble features.
- Calling the **EnsLib.HL7.Util.Generator** method to generate the XML representation of the schema and then explicitly accessing the component length attribute or the data structures component. Specifically, the component length was specified by the `length` attribute in previous versions but is specified by the `max_length` attribute in Ensemble 2014.1. The same restriction applies if you generate the XML schema by exporting from the portal or from Studio.
- Calling the `EnsLib.HL7.Schema.DataStructures` query. This query returns additional information, specifically, information about the elementary data types. If your code is dependent on the data returned by this query, there may be compatibility issues.
- Exporting an HL7 schema in Ensemble 2014.1 or later and then attempting to import it into an earlier version of Ensemble. The earlier version of Ensemble will encounter errors caused by the new segment length attributes and component data structure. You can export an HL7 schema using an older version of Ensemble and then import it into Ensemble 2014.1 or later.

8

Ensemble 2013.1

8.1 New Features in Ensemble 2013.1

Ensemble 2013.1 introduced the following new features and enhancements:

- Complex Record Mapper
- Enterprise Message Viewer
- Custom HL7 Schema Editor
- Temporary variables in business rules
- Enhanced EDI XML supports Document.Validate method
- Documentation reorganization and MSMQ Adapter documentation removed

8.2 Compatibility Issues for Upgrades to Ensemble 2013.1

The following changes in this release may affect the operation of your existing system. Review these following issues before upgrading a previous instance of Ensemble.

- [Simplified X12 Schema Representation](#)
- [Need to Perform Cleanup after Upgrading from Previous Versions](#)
- [Improve Access to ASTM Documents by Building Index](#)
- [Enable Access to DeepSee Portal Pages](#)

Also review the *Caché Release Note and Upgrade Checklist Archive*.

8.2.1 Simplified X12 Schema Representation

In this release, when you load a SEF file into Ensemble, Ensemble converts the schemas into a format that is simpler than the raw SEF definition. This simplified X12 schema format can be exported as XML and edited in Studio. The main simplifications of the schema format are as follows:

- Segment definitions are the same across a schema, rather than different for every usage of a segment within a structure within a schema.

- Successive segments of the same type are rolled into one repeating segment.
- Loops are qualified using key fields for HL or NK1 trigger segments only where needed for disambiguation.

It is necessary to adjust any X12 virtual property paths that you created in Ensemble version 2012.1 or earlier.

8.2.2 Need to Perform Cleanup after Upgrading from Previous Versions

If you upgraded Ensemble from a previous version to version 2012.2, 2012.2.1, or 2012.2.2, the upgrade process cleaned up the UTC index. In some installations, this cleanup process took an extended time and delayed the upgrade process. In Ensemble 2012.2.3, Ensemble 2013.1, and later versions, this cleanup process is not automatically done during the upgrade but should be done manually after the upgrade. If you have already upgraded a system to Ensemble 2012.2.1 or 2012.2.2, the cleanup process has been completed and you do not need to do anything for any future upgrades.

If you are upgrading from Ensemble 2009.1 or earlier, some message headers or log entries won't be visible until after you run the cleanup procedure. Consequently, you should run the cleanup procedure soon after upgrading Ensemble. If you are upgrading from Ensemble 2010.1 or later, you can run the cleanup procedure at a convenient time after upgrading Ensemble.

If you are upgrading from Ensemble 2012.1 or earlier to Ensemble 2012.2.3, Ensemble 2013.1, or a later version, follow this procedure:

1. When Ensemble is being upgraded, it detects whether any namespaces on the system need the UTC index cleanup. If any namespaces need the cleanup, Ensemble logs a warning to `cconsole.log` indicating each namespace that needs to be cleaned up.
2. Check the `cconsole.log` file after upgrading to find the namespaces that need to be cleaned up.
3. After the system has been upgraded and Ensemble is running normally, you can run the cleanup on each namespace that requires it. You may wish to run the cleanup at an off-peak time when the system load is low. To run the cleanup for a specified namespace, enter the following in Ensemble Terminal:

```
zn <namespace>  
Do ##class(%Library.EnsembleMgr).UpgradeUTCIndices()
```

You can also check to see if a namespace requires the UTC index cleanup by entering the following at the terminal:

```
Do ##class(%Library.EnsembleMgr).CheckUTCIndices(<namespace>,1,1)
```

If the namespace needs cleanup, `CheckUTCIndices` displays a message indicating that cleanup is required for the specified namespace. Otherwise it displays a message indicating that no UTC index globals need repair.

8.2.3 Improve Access to ASTM Documents by Building Index

Ensemble 2012.1 and later versions provide improved access to ASTM documents by adding an index for the `OriginalDocId` field. If you have created your production with Ensemble 2012.1 or later, this index is present for all ASTM documents. However, if you created your production with Ensemble 2010.2 or an earlier version and then have upgraded to 2012.1 or a later version, the index is created for new documents, but any existing documents created with the earlier version will not have the `OriginalDocId` index.

After upgrading Ensemble, if you have ASTM documents, you should build the index for the `OriginalDocId` field. You can build the index while the production is running, but you cannot build the index while ASTM messages are being purged or deleted. To rebuild the index, enter the following in Ensemble Terminal for each namespace that contains ASTM documents:

```
zn <namespace>  
Do ##class(EnsLib.EDI.ASTM.Document).%BuildIndices("OriginalDocId")
```

For information on ASTM documents, see the [Ensemble ASTM Development Guide](#).

8.2.4 Enable Access to DeepSee Dashboard Pages

Beginning in this release, application access to arbitrary %CSP pages, including DeepSee, can now be better controlled. By default, only the SAMPLES namespace can access DeepSee pages, including dashboards. To enable DeepSee access in the ENSDEMO namespace and its associated web application, /csp/ensdemo, enter the following command in an Ensemble terminal window:

```
Do EnableDeepSee^%SYS.cspServer("/csp/ensdemo/")
```

You should enter this command for any namespace that uses DeepSee dashboards or other DeepSee pages. In order to find the name of the web application associated with an Ensemble namespace, go to the **[System Administration] > [Security] > [Applications] > [Web Application]** page in the Ensemble management portal. Note that for HealthShare installations, the web application name is /csp/healthshare/ensdemo. Alternatively, you can enable DeepSee access for all namespaces and web applications by entering the following command:

```
Do EnableDeepSee^%SYS.cspServer(0)
```

For a detailed description of this issue, see [Application Access To %CSP Pages Now Controlled for 2013.1](#) in the *Caché Release Note and Upgrade Checklist Archive*.

9

Ensemble 2012.2

9.1 New Features in Ensemble 2012.2

Ensemble 2012.2 introduced the following new features and enhancements:

- SAP JCo Adapter
- NHS Interoperability Toolkit (ITK)
- .NET 4.0 Components for Object Provider and .NET Gateway
- Custom Search Table Classes for Virtual Documents
- Custom Queries for Messages
- Ability to Bind to Specific Network Address in TCP Adapters
- Better Handling of Large HL7 Messages
- Easier Control Over Saving HL7 Reply Messages
- XML Virtual Document Enhancements

9.2 Compatibility Issues for Upgrades to Ensemble 2012.2

The following changes in this release may affect the operation of your existing system. Review these following issues before upgrading a previous instance of Ensemble.

Also review the *Caché Release Note and Upgrade Checklist Archive*.

9.2.1 New Compiler Behavior for DTL Classes

In previous releases, the system threw an exception when you compiled a DTL and the source or target class does not exist. This means that your compiled DTL code was at risk of being incorrect. Now the compiler reports an error and fails if the source or target class does not exist.

9.2.2 New DTL Classes Created with IGNOREMISSINGSOURCE Parameter Set to True

Previous to this release, all DTL data transformation classes inherited from `Ens.DataTransformDTL` had the `IGNOREMISSINGSOURCE` parameter set to `False`. Beginning with this release, any DTL classes you create with the Data Transform wizards in the Management Portal or Studio override this value to `True` with the following declaration:

```
Parameter IGNOREMISSINGSOURCE = 1;
```

With this parameter value, the DTL suppresses errors caused by attempts to get field values out of absent source segments. The DTL also skips calling subtransforms where the named source segment is absent.

However, to maintain compatibility for existing DTL data transformation classes, the default behavior of the abstract class has not changed, and your existing DTL classes behave as in the past. The `Ens.DataTransformDTL` class declares the following:

```
Parameter IGNOREMISSINGSOURCE = 0;
```

You can review your transformations to see if updating the value of this parameter makes sense in your application.

9.2.3 Rules Conversion and Upgrade

During an upgrade to Ensemble 2012.1 or later, Ensemble ensures that any existing business rules are correctly converted and upgraded. As part of this conversion, Ensemble clears the custom function cache before performing the conversion. The cache is cleared initially to ensure that the correct resolution is performed while converting the rules.

In addition, the upgrade code compiles any custom `FunctionSet` classes (subclasses of `Ens.Rule.FunctionSet`) that need to be recompiled *before* converting and compiling rules created using the rules engine in 2010.2 and older. You should ensure that any mapped `FunctionSet` classes can be compiled in their target namespaces.

9.2.4 Change in Operator Precedence in Business Rules and Routing Rule Conditions

In release 2012.1, the precedence of conditions in rule conditions was not always correct. In particular, comparison operators took precedence over arithmetic and concatenation operators, so an expression of the form `'a+b=c'` was being evaluated as `'a+(b=c)'`. In 2012.2 this is correctly evaluated as `'(a+b)=c'`.

If you have written rules in 2012.1 that relied upon this incorrect behavior, those rules will no longer function correctly and the rules will have to be changed. Note that the Visual Rule editor adds parentheses to conditions and there is no problem.

Conditions in rules upgraded directly from 2010.2 or earlier to 2012.2 will execute correctly.

Conditions in rules upgraded from 2010.2 or earlier to 2012.1 may not execute correctly in that version. After a subsequent upgrade to 2012.2 these rules will once again execute correctly.

10

Ensemble 2012.1

10.1 New Features in Ensemble 2012.1

Ensemble 2012.1 introduced the following new features:

- Redesigned User Interface
- Granular Security in the Management Portal
- New Business Rules
- Source Control Hooks in the Management Portal
- Record Mapper
- Record Batch Handling
- XML Virtual Documents
- Alert Generation on Monitored Thresholds
- UDDI API
- Caché 2012.1 Features

Ensemble 2012.1 provided enhancements to the following features:

- DeepSee Style Dashboards
- ebXML Messaging
- Sequence Manager
- Creating a Studio Project from a Production

10.2 Compatibility Issues for Upgrades to Ensemble 2012.1

The following changes in this release may affect the operation of your existing system. Review the following issues before upgrading a previous instance of Ensemble:

- [New Management Portal User Interface](#)
- [Business Rule Conversion](#)
- [Changes in Rule Log Structure](#)
- [New Dashboard Development Tool](#)
- [New Security Model for Management Portal](#)
- [Changes to Workflow User Interface](#)
- [Updated Selectivity and Extent Size of the Message Warehouse](#)
- [Updated Saved Message Searches](#)
- [Removed CSPX Files from Distribution](#)
- [Updated Search Table Validation](#)
- [New DTL Classes Created with REPORTERRORS Parameter Set to True](#)
- [Updated Legal Character Checking in Configuration Names](#)
- [Change in Inactivity Timeout Behavior](#)
- [Removed Host Monitor from User Interface](#)
- [Improved Notification for Stopping a Running Production](#)
- [Change in \\$\\$\\$EnsSystemError Behavior](#)

Also review the *Caché Release Note and Upgrade Checklist Archive*.

10.2.1 New Management Portal User Interface

The user interface for the Ensemble Management Portal is completely new in this release; therefore, any procedures you are using or have documented most likely must change. Each page of the new portal has help information to guide you.

See *Managing Ensemble* for details.

10.2.2 Business Rule Conversion

The upgrade procedures to this release automatically convert existing rules. Old rule names (with a .rul extension) allowed for characters that are not supported in class names. In this case, the rule definition includes an alias that is used to invoke the rule. Old rules are automatically converted on upgrade or import and the changes should not affect your applications. However, if you rely on details of the old implementation, you may encounter issues.

Due to the more limited structure of the old rules, some converted rules end up with a structure that may not be the most straightforward or recommended way of developing rules in the new editor, but they will still work as before.

For HL7 routing rules, the rule editor no longer exposes the *Schema DocType*, which represents the message structure. This is now inferred from the message type. The message structure continues to be exposed for rules that you import from earlier versions.

10.2.3 Changes in Rule Log Structure

As part of the changes to the implementation of business rules and routing rules, this release changes the structure of the rule log. The previous rule log is still available in the `Ens.Rule.RuleLog` class (SQL table `Ens_Rule.RuleLog`). The new rule log is stored in `Ens.Rule.Log` (SQL table `Ens_Rule.Log`).

The business rule conversion described in the previous section now records slightly different information in the rule log, in addition to using a different storage structure. If you have written your own queries or reports based on the contents of the rule log, you must update your queries to ensure that they continue to retrieve the correct information and that they continue to perform optimally.

Important: You cannot view business rule log entries created prior to an upgrade to Ensemble 2012.1 on the **[Ensemble] > [Rule Log]** page in the new Management Portal.

The **Rule Log** page only shows entries in the new rule log. Ensemble does, however, provide a legacy page, **[Ensemble] > [Legacy Rule Log]**, so you can see the rule log entries from an earlier version:

`http://localhost:57772/csp/ensemble/EnsPortal.LegacyRuleLog.zen`

You cannot navigate to this legacy page from the Management Portal menus; you must enter the above URI, replacing 57772 with the web server port of your Ensemble instance. The `EnsPortal.LegacyRuleLog` page is subject to the same security restrictions as the `EnsPortal.RuleLog` page.

Ensemble purges the legacy log in the usual way; all old entries should remain for only a small number of weeks depending on your retention policy.

10.2.4 New Dashboard Development Tool

Existing dashboards are not operational starting with this release, but existing business metrics are still valid. You cannot directly convert dashboards from previous releases to this release. Instead, you must create a dashboard in the DeepSee User Portal using your existing business metric as the data source to implement each new dashboard. If you require this type of update, contact the [InterSystems WRC](#) for guidance.

See the chapter “[Creating Dashboards](#)” in *Configuring Ensemble Productions* to get started.

10.2.5 New Security Model for Management Portal

If you are upgrading an instance of Ensemble, the upgrade process adds new roles to Ensemble users based on their previous roles. Review these roles after an upgrade to verify the converted access and to further restrict access as needed.

One exception is that users in previous versions who held the `%Service_Login` resource were able to start or stop productions from the command line, even if they did not have permission to access the Management Portal. After an Ensemble upgrade, these users will not be able to stop or start productions. To allow them to do so, you must give them a role, such as `%Ens_Operator`, that holds the required resource. Users who could start or stop productions from the Ensemble Management Portal in earlier releases are not affected by this tightening in security checks.

For more information, see “[Controlling Access to Management Portal Functions](#)” in *Managing Ensemble*.

10.2.6 Changes to Workflow User Interface

As of this release, the separate Workflow Management Portal is gone. Instead, Ensemble now provides two user interfaces, intended for different sets of users:

- The Management Portal provides pages that implementers and supervisors can use to manage workflow roles, users, and tasks. To access them, from the **Ensemble** menu, click **Manage** and then click **Workflow**.

These pages are similar in most ways to the Workflow Management Portal, except that they no longer provide options for users to manage their tasks.

- The DeepSee User Portal is a stand-alone user interface intended for end users. It displays dashboards (both Ensemble and DeepSee dashboards). For workflow users, it includes the Workflow Inbox. Via the Workflow Inbox, users can display the task form associated with a task and manage that task in the same way as in previous releases.

10.2.7 Updated Selectivity and Extent Size of the Message Warehouse

Ensemble uses the extent size and selectivity (the number or percentage of records that match any value) property parameters in any table to optimize SQL queries. Setting these correctly for `Ens.MessageHeader` is important for good response times when using the Ensemble message browser. You can set these values by running the Tune Table facility against this class.

The Ensemble upgrade procedure, however, overwrites this information with the default values. This release improves the default values to represent a typical large site, which should give good performance for most installations. The following shows the default *ExtentSize* and *Selectivity* values for the `Ens.MessageHeader` class in this Ensemble release:

```
<ExtentSize>20000000</ExtentSize>
<IdLocation>^Ens.MessageHeaderD</IdLocation>
<IndexLocation>^Ens.MessageHeaderI</IndexLocation>
<Property name="MessageBodyClassName">
<Selectivity>10%</Selectivity>
</Property>
<Property name="MessageBodyId">
<Selectivity>0.0001%</Selectivity>
</Property>
<Property name="Priority">
<Selectivity>20%</Selectivity>
</Property>
<Property name="SessionId">
<Selectivity>20%</Selectivity>
</Property>
<Property name="SourceConfigName">
<Selectivity>5%</Selectivity>
</Property>
<Property name="Status">
<Selectivity>11%</Selectivity>
</Property>
<Property name="TargetConfigName">
<Selectivity>5%</Selectivity>
</Property>
<Property name="TimeCreated">
<Selectivity>5%</Selectivity>
</Property>
<Property name="TimeProcessed">
<Selectivity>5%</Selectivity>
</Property>
```

If you have a large message warehouse and have either run Tune Table or have set these values manually, you should either verify that the default values match your existing system or run Tune Table after the upgrade. If you have not run Tune Table or manually set the *Selectivity* value, the new values should improve the performance of queries in the Message Viewer.

If you have taken action to optimize access to this table, take the following actions to ensure that the system performs well after the upgrade:

1. Record the *ExtentSize* and *Selectivity* values of your current system. One way to do this is to open the `Ens.MessageHeader` class in Studio. The *ExtentSize* and *Selectivity* values are listed near the end of the class definition in the code window.
2. If your values are significantly different than the new defaults, then after upgrading, either run Tune Table or use Studio or the Management Portal to manually update the *ExtentSize* and *Selectivity* values to describe your system.

Important: You can run Tune Table against a running system as long as you select the **Keep class up to date** check box.

For details on using Tune Table from the **[System] > [SQL] > [Schemas] > [Tables]** page of the Management Portal, see “ExtentSize, Selectivity, and BlockCount” in the chapter “Optimizing Tables” in the *Caché SQL Optimization Guide*.

10.2.8 Updated Saved Message Searches

In Ensemble, *saved message filters* or *saved searches* allow you to give a name to frequently used combinations of criteria in the message viewer. The storage of these filters has changed and during an upgrade the message filters are automatically

converted to the new format and no action is required. However, if you want to export filters from an earlier release to 2012.1 you must run a manual conversion.

First, export the global `^CSPX.EnsMsgFilterFormD` from the earlier version and import it into your Ensemble 2012.2 instance.

Then either convert a single saved search with the following command:

```
Do ##class(EnsPortal.MsgFilter.SavedSearch).ConvertCSPXSearch("mysearch")
```

Or, convert all saved searches with the following command:

```
Do ##CLASS(EnsPortal.MsgFilter.SavedSearch).ConvertAllCSPXSearches()
```

10.2.9 Removed CSPX Files from Distribution

Ensemble no longer ships CSPX files as part of the distribution. The inclusion of these files would give people the ability to bypass the granular security of the new user interface by accessing the old user interface. If you previously used these files, contact the [InterSystems WRC](#) for help in upgrading your Ensemble environment.

10.2.10 Updated Search Table Validation

An update to the consistency checking, storage definition, and upgrade procedure for [search tables](#) resolves a long-standing issue with upgrades losing search table metadata. Ensemble now stores search table metadata locally in each namespace.

To avoid rebuilding your custom search tables after an upgrade, perform an additional step (step 4 of the [Upgrading Ensemble](#) procedure) before upgrading to ensure that Ensemble correctly retains search table metadata.

Search table metadata is located in the default global database for each Ensemble namespace; therefore, a change to a search table class does not update metadata in all namespaces to which the class is mapped. You must compile a mapped search table class in all target namespaces to ensure that the metadata local to each namespace is up to date.

Note: After the upgrade to this release, your existing search tables contain updated metadata in the appropriate namespaces; you do not need to recompile them. However, you must follow the described compile procedure for any search tables you add or change.

If you have not developed any custom search tables, you do not need to take any action. If you complete the upgrade without performing the additional pre-upgrade step and then determine you do have custom search tables, values in the search tables may be incorrect. You can correct this by rebuilding the search tables. For each search table, perform the following:

```
Set sc=##class(EnsLib.HL7.SearchTable).BuildIndex()
```

See the `EnsLib.HL7.SearchTable` entry in the *Class Reference* for details.

Note: Running the `EnsLib.HL7.SearchTable.BuildIndex()` class method generates journal entries and could take time. You can run it while messages are processing and you can run it in batches specifying a start and end ID. You do not need to include messages processed since the upgrade.

10.2.11 New DTL Classes Created with REPORTERERRORS Parameter Set to True

Previous to this release, all DTL data transformation classes inherited from `Ens.DataTransformDTL` had the `REPORTERERRORS` parameter set to `False`. Beginning with this release, any DTL classes you create with the Data Transform wizards in the Management Portal or Studio override this value to `True` with the following declaration:

```
Parameter REPORTERRORS = 1;
```

This setting causes Ensemble to log any errors it encounters in executing the transform as *Warnings* in the Event Log and to return a composite status code containing all errors as its return value.

However, to maintain compatibility for existing DTL data transformation classes, the default setting in the abstract class did not change. The `Ens.DataTransformDTL` class still declares:

```
Parameter REPORTERRORS = 0;
```

This setting causes Ensemble to silently log errors as trace messages with category `xform`.

You can review your transformations to see if updating the value of this parameter makes sense in your application.

10.2.12 Updated Legal Character Checking in Configuration Names

The `[` character is now disallowed in production configuration names. Productions containing configuration items with names that contain this character no longer compile successfully. As described in the `CheckForIllegalCharacters()` method of the `Ens.Config.Item` entry of the *Class Reference*. This character is restricted because it could interfere with the `ArchiveItems` property setting syntax of the Message Bank operation (`Ens.Enterprise.MsgBankOperation`) class.

10.2.13 Change in Inactivity Timeout Behavior

The behavior of the `InactivityTimeout` setting now sends an alert in addition to marking a component as *Inactive* when no activity has occurred within the inactivity timeout of a configuration item. In addition, the setting is included in the `Settings` property of the `Ens.Config.Item` class to permit the use of Default Settings to populate this value.

The original `InactivityTimeout` property of the `Ens.Config.Item` class and the XML attribute of the same name is transparently transferred to the new location, so previous code directly accessing this value should see no change in behavior, but the structure of the XML produced in the production XData is slightly different.

10.2.14 Removed Host Monitor from User Interface

Previous releases of Ensemble provided a **Host Monitor** page in the Ensemble Management Portal. The new Management Portal user interface does not contain this specific page, but does contain the following pages available from the **Monitor** menu in the Ensemble portion of the portal for monitoring your Ensemble productions:

- **System Monitor**
- **Production Monitor**
- **Queues**
- **Jobs**

See the [Monitoring Ensemble](#) book for details.

10.2.15 Improved Notification for Stopping a Running Production

In the new Management Portal user interface, you stop a production from the `[Ensemble] > [Production Configuration]` page. You can only stop a production if it is open for configuration and it is running. You receive an informational message if you try to stop a production that is not running or try to start a production and another production is already running in the namespace. You must open the running production in the **Production Configuration** page before you can stop it.

In previous Ensemble versions, if one production was running and you were configuring a different production, when you clicked **Stop Production** on the configuration page, Ensemble would stop the running production regardless if it was open for configuring. This could lead to a user inadvertently stopping the wrong production.

You can view the running productions on the right hand side of the Management Portal menu navigation pages and click **View details** to open the selected production in the **Production Configuration** page.

10.2.16 Change in \$\$\$EnsSystemError Behavior

In previous releases, the \$\$\$EnsSystemError macro logged all exceptions it trapped in Ensemble to the %ETN utility. This release has updated this behavior; it makes the logging optional and turned off by default.

The logging is now controlled by the `^Ens.Debug("LogETN")` global. This global is undefined by default, so %ETN logging does not occur. You can set the global at any time to a nonzero value to enable %ETN logging. The purpose of this change is to avoid consuming excessive database space when repetitive errors occur in an Ensemble production. Allowing it to be enabled by setting a global means it can be turned on at any time to collect deeper information when a problem is occurring.

If you have used the \$\$\$EnsSystemError macro to log exceptions to %ETN in your application, you must set the `^Ens.Debug("LogETN")` global for your application error logging to continue.

10.3 Finding Ensemble Options in the New Management Portal

For users who are familiar with the Ensemble Management Portal in 2010.x, this appendix describes where to find Ensemble options in the combined Management Portal in 2012.x.

The following table lists options in the 2010.x Ensemble Management Portal and indicates at least one way to find the option in the new portal. Note that this table indicates the paths needed to find the options, and these paths are not always the same as the breadcrumbs shown at the tops of the pages.

Option in the 2010.2.x Ensemble Management Portal	Location in the 2012.x Combined Management Portal
Home link	Ensemble > Monitor > Production Monitor (this page has been redesigned extensively)
Productions link	Ensemble > List > Productions menu option
Production Configuration page (available from the Home link or the Productions link)	Ensemble > Configure > Production and Ensemble > View > Production menu options
System Monitor link	Ensemble > Monitor > System Monitor menu option (this page has been redesigned extensively)
Event Log link	Ensemble > View > Event Log menu option
Message Browser link	Ensemble > View > Messages menu option (no longer provides an option to compare messages)
Visual Trace page (available from the Message Browser link)	Visual Trace page (available from the Ensemble > View > Messages menu option and other locations)
Business Processes link	Ensemble > List > Business Processes
BPL Viewer page (available from the Business Processes link)	Ensemble > Build > Business Processes
Business Rules link	Ensemble > List > Business Rules

Option in the 2010.2.x Ensemble Management Portal	Location in the 2012.x Combined Management Portal
Ensemble Business Rule Editor (available from the Business Rules link)	Ensemble > Build > Business Rules
Business Rule Log link	Ensemble > View > Business Rule Log
Host Monitor link	<i>Not available</i>
Jobs link	Ensemble > Monitor > Jobs menu option
Queues link	Ensemble > Monitor > Queues menu option
Maintenance link	Ensemble > Manage > Auto-Start Production menu option
Maintenance > Suspended Messages link	Ensemble > View > Suspended Messages menu option
Maintenance > Purge Management Data link	Ensemble > Manage > Purge Management Data menu option
Maintenance > Credentials link	Ensemble > Configure > Credentials menu option
Maintenance > Local Archive Manager link	Ensemble > Manage > Local Archive Manager menu option
Maintenance > Lookup Settings link	Ensemble > Configure > Data Lookup Tables menu option
Maintenance > DICOM Settings link	Ensemble > Interoperate > DICOM menu
Maintenance > Default Settings link	Ensemble > Configure > System Default Settings menu option
Maintenance > Message Bank link	Ensemble > Configure > Message Bank Link menu option
Dashboards link	DeepSee > User Portal menu option
Business Metrics link	<i>Not available</i>
Workflow Portal > My Tasks link	DeepSee > User Portal menu option (the User Portal displays a folder of your workflow tasks)
Workflow Portal other links	Ensemble > Manage > Workflow menu
Testing Service link	Ensemble > Test > Business Hosts menu option
EDI / HL7 Manager > ASC X12 options	Ensemble > Interoperate > ASC X12 menu
EDI / HL7 Manager > ASTM options	Ensemble > Interoperate > ASTM menu
EDI / HL7 Manager > HL7 version 2 options	Ensemble > Interoperate > HL7 v 2.x menu
EDI / HL7 Manager > UN/EDIFACT options	Ensemble > Interoperate > UN/EDIFACT menu

11

Ensemble 2010.2

11.1 New Features in Ensemble 2010.2

Ensemble 2010.2 introduced the following new features:

- DICOM Support
- Enterprise Monitor
- Enterprise Message Bank
- SFTP Support
- Ensemble and Mirroring
- Caché 2010.2 Features

Ensemble 2010.2 provided enhancements to the following features:

- Visual Trace
- Improved HL7 and X12 Host Wizards
- Large Object (LOB) Support of Input Parameters for the SQL Outbound Adapter
- SSL/TLS Support on Inbound (POP3) and Outbound (SMTP) Email Adapters
- Additional Configuration Settings for Inbound Adapters

11.2 Compatibility Issues for Upgrades to Ensemble 2010.2

The following changes in the 2010.2 release may affect the operation of your existing system. Review the following issues before upgrading a previous instance of Ensemble:

- [Remove Support for HL7v2 Framing with XML Text](#)
- [Update Error Handling on HTTP Outbound Adapter](#)
- [Update Error Processing in File Outbound Adapter](#)

- [Change Return Status on HTTP Inbound Adapter](#)
- [Add Requirement to Subclass Message Bank Production](#)
- [Update Disable Behavior of Business Processes](#)

Also review the *Caché Release Note and Upgrade Checklist Archive*.

11.2.1 Remove Support for HL7v2 Framing with XML Text

This release of Ensemble removes support for all HL7 framing options that involve XML text being detected or generated in between successive HL7 message bodies in an HL7 data stream. This is an undocumented feature InterSystems believes no one is using. If you are using any of these options, contact the [InterSystems WRC](#).

11.2.2 Update Error Handling on HTTP Outbound Adapter

This release of Ensemble updates HTTP outbound adapter processing to return an error status code (`<Ens>ErrHTTPStatus`) if the HTTP status it receives is something other than 200 (OK). Also, the adapter now sets the retry flag if it receives a status of 503 (Service unavailable due to a temporary overloading or maintenance of the server). The introduction of the new status code makes error handling more accessible to the **Reply Code Actions** setting feature. See “[Reply Code Actions](#)” in the reference section of *Configuring Ensemble Productions*.

This change also updates the HL7 HTTP outbound adapter to return the indicated ACK commit code according to the HTTP status conditions shown in the following table.

ACK commit code	HTTP status condition
AA	200 — OK code
AR	503 — Service Unavailable due to a temporary overloading or maintenance of the server
AE	All other non-OK codes

See “[HL7 Acknowledgement \(ACK\) Mode](#)” in the *Ensemble HL7 Version 2 Development Guide* for more information.

If you have code that expects a `$$$OK` status returned from methods of the HTTP outbound adapter even when the remote HTTP server returns a non-OK status, you may need to update the code to either change the error handling or configure the **Reply Code Actions** setting to recognize the new error code.

11.2.3 Update Error Processing in File Outbound Adapter

This release of Ensemble improves the error status checking and error trapping in the `PutStream()` method of the `EnsLib.File.OutboundAdapter`.

11.2.4 Change Return Status on HTTP Inbound Adapter

This release of Ensemble changes the HTTP inbound adapter return status to a server error instead of OK if the `ProcessInput()` method returns an error status.

If you have clients invoking an Ensemble service that uses the HTTP inbound adapter you may now see an HTTP error status code (500) when an error occurs in the Ensemble service, when formerly you saw an HTTP OK (200) status. This does not disrupt normal operation because it only affects behavior when the HTTP service fails. Additionally, Ensemble still returns its non-standard `<error>` block body. It is unlikely that your service has customized behavior based on this

returned status; however, this change may trigger a different code path in your error handler and therefore you should review this code.

11.2.5 Add Requirement to Subclass Message Bank Production

This release of Ensemble changes the `Ens.Enterprise.MsgBank.Production` class to be an abstract class and adds a requirement that you must subclass it and copy the `ProductionDefinition` XData block, to run a Message Bank instance. This allows you to run multiple message banks in separate namespaces on the same instance, and it prevents future upgrades from deleting your configuration setting changes. It also removes an obstacle to allowing you to mark your ENSLIB database as read-only.

If you are an early adopter of the Message Bank from a previous release, you must copy your Message Bank production class (`Ens.Enterprise.MsgBank.Production`) to a subclass before upgrading. If you do not, the upgrade will overlay your configuration changes, and will not allow you to restart the common Message Bank production or reapply your configuration settings.

11.2.6 Update Disable Behavior of Business Processes

This release of Ensemble refines the behavior of disabling a business process. The behavior depends on the private **Pool Size** configuration setting of the business process:

- **Business process Pool Size > 0:**

The business process only uses jobs from its private pool; you can disable just this process by clearing the **Enabled** check box on the configuration page of the business process.

- **Business process Pool Size = 0:**

The business process shares the public actor pool job queue (**Actor Pool Size**) with all other business processes with a **Pool Size = 0**. Disabling one such business process, disables the `Ens.Actor` queue, effectively disabling all business processes that use the actor pool. If you clear the **Enabled** check box of a business process that has a **Pool Size = 0**, you receive the following message:

```
WARNING: 'Enabled' is not checked and 'PoolSize' is 0. If you save these
settings, the Ens.Actor shared actor queue will become disabled,
effectively disabling all other business processes that also use the shared
queue. If this is not what you want, you can still disable this business
process, but first set 'PoolSize' > 0 so that this business process uses its
own dedicated queue. Then you can safely disable it.
```

```
Are you sure you want to disable all business processes?
```

If you upgrade to this release and your production contains a business process with **Pool Size = 0**, disabling the process now has different behavior.

For a detailed discussion of pool sizes, see “[Pool Size](#)” in *Configuring Ensemble*.

12

Ensemble 2010.1

12.1 New Features in Ensemble 2010.1

Ensemble 2010.1 introduced the following new features:

- Configuration Default Settings
- EDIFACT Support
- Caché 2010.1 Features

12.2 Compatibility Issues for Upgrades to Ensemble 2010.1

The following changes in the 2010.1 release may affect the operation of your existing system. Review the following issues before upgrading a previous instance of Ensemble:

- [Relocate RemoveItem\(\) Configuration Method](#)
- [Add Configuration Setting on TCP Outbound Adapter](#)
- [Correct Type Node in HL7 Sequence Manager Global](#)
- [Correct Behavior of HL7 Configuration Framing Setting](#)
- [Add Support for Legacy FTPS Protocol to FTP Adapters](#)
- [Changes in Mapping of Custom Schemas](#)

Also review the *Caché Release Note and Upgrade Checklist Archive*.

12.2.1 Relocate RemoveItem() Configuration Method

In this release, Ensemble moves the **RemoveItem()** method from the CSPX.EnsConfigProperty class to the Ens.Config.Production class; it is now available for general use and not exclusively from the Ensemble Management Portal configuration page.

Calls to the undocumented **CSPX.EnsConfigProperty.RemoveItem()** method in your code, receive a <METHOD NOT FOUND> error. In the unlikely event you use this method, update your code to now use the **Ens.Config.Production.RemoveItem()** method. Instead of passing your production object as an argument, the new method is an instance method of your production object.

12.2.2 Add Configuration Setting on TCP Counted Outbound Adapter

This release adds a new `FlushBeforeSend` configuration setting to the TCP Counted Outbound Adapter. When set to `True`, this option causes the `SendMessageStream()` adapter method to do a zero-timeout read of all data pending in the inbound TCP buffer before writing its outbound data and optionally reading any subsequent returning data.

If you had implemented a block protocol using the TCP Counted Outbound Adapter in previous releases of Ensemble, you must override the default setting.

12.2.3 Correct Type Node in HL7 Sequence Manager Global

In previous releases, the Type subscript `^EnsHL7.SM("output",type)` incorrectly used `PerformOutputTransformationOn`. The class documentation for `EnsLib.HL7.SM.RuntimeData` has been updated. See the entry in the *Class Reference* for details.

Existing applications will encounter problems if they have `PerformOutputTransformationOn` set to `SequenceNumberOnly`. If so, and you want to keep the existing output sequence number, perform the following:

```
Merge ^EnsLib.SM("output", "Sender") = ^EnsLib.SM("output", "SequenceNumberOnly")
```

Also verify that your `PerformOutputTransformationOn` and `OutputSequenceNumberIndexField` are consistent.

12.2.4 Correct Behavior of HL7 Configuration Framing Setting

This release corrects the behavior of HL7 business services and business operations when you configure the *Framing* setting to have a value of `None`. This value now results in no framing characters being generated between HL7 messages as opposed to the previous behavior that used whatever framing was declared as the default in the relevant context.

Productions configured with `Framing=None` for various configuration items may be experiencing incorrect framing behavior that works in your context. This change corrects the behavior which may cause your production to stop working. For example, you may be sending outbound files to an entity expecting an ASCII LF between messages; even though the file operation is configured to put nothing between messages because previously it had been erroneously generating the LF between messages.

12.2.5 Add Support for Legacy FTPS Protocol to FTP Adapters

Release 2009.1 of Caché implemented the RFC4217 standard method of creating a secure FTP transfer, and it also removed the previous legacy mode which assumed that the command channel was to use TLS. However, some Ensemble implementations using FTP adapters were using this mode. The current release reintroduces this legacy connection mode with a special way in the FTP adapter configuration to indicate its use.

If you have been using the old non-standard FTPS protocol first implemented in the `%Net.FtpSession` class, you may find that your FTP adapters no longer work with the FTP servers to which they have been connecting. To restore proper functioning of the adapter, append an asterisk (*) to the `SSLConfig` property of the appropriate `EnsLib.FTP.InboundAdapter` or `EnsLib.FTP.OutboundAdapter` class.

See the `SSLConfig` property description in the `EnsLib.FTP.Common` entry of the *Class Reference* for details.

12.2.6 Changes in Mapping of Custom Schemas

In previous releases, your custom HL7 and EDI schemas were stored in the ENSLIB namespace; therefore, they were mapped to every namespace. However, the Ensemble upgrade procedure replaces everything in the ENSLIB namespace, so you would have to export and then import your defined schemas to save them when you upgraded.

Beginning with Ensemble 2007.1, only the standard schemas are available in all namespaces. Ensemble now stores all custom HL7 and EDI schemas in the namespace where you define them. If you depended on centrally located schemas in your previous Ensemble version, you must now compile your user-defined schemas in each namespace where you use them.

13

Ensemble 2009.1

13.1 New Features in Ensemble 2009.1

Ensemble 2009.1 introduced the following new features:

- Ability to Edit and Resend Messages
- Object Gateway for .NET Services and Operations
- Ability to Create a Studio Project from a Production
- Automatic Documentation of a Production
- Support for ebXML
- Caché 2009.1 Features

Ensemble 2009.1 provided enhancements to the following features:

- Testing Service for Virtual Document Messages
- Sequence Manager Support for Non-HL7 Messages
- SQL Adapter
- Java Gateway Adapters
- Additional HL7 Schema Definitions
- Visual Trace
- Ensemble Automatic Start Setting

13.2 Compatibility Issues for Upgrades to Ensemble 2009.1

The following changes in the 2009.1 release may affect the operation of your existing system. Review the following issues before upgrading a previous instance of Ensemble:

- [Changes in HL7 Storage Structure](#)

- [New ReplyCodeActions Property in Process and Operation Classes](#)
- [New Mechanism for Editing Messages Replaces the %DrawEditForm\(\) Method](#)
- [Increased Alert Level for Data Transformation Errors](#)
- [Changes to Pool Size Configuration Behavior on TCP Service](#)
- [Renamed Column in Statistics Queries](#)
- [Alert Support for Services Invoked Outside Ensemble](#)
- [Changes in Empty Schema Category Behavior](#)

Also review the *Caché Release Note and Upgrade Checklist Archive*.

13.2.1 Changes in HL7 Storage Structure

This release of Ensemble changes the storage structure for HL7 message segments to avoid block contention and improve throughput of large systems.

Ensemble now stores message segments in the new format and converts old message segments to the new format the first time it opens the message as an object. Access to HL7 messages from SQL and from the Management Portal is compatible with both formats.

This change is transparent to most applications; however, if you have code that directly accesses or manipulates the segment globals, you must modify it to be compatible with the new structure. Contact the [InterSystems WRC](#) for advice and guidance if you need to make such changes.

13.2.2 New ReplyCodeActions Property in Process and Operation Classes

This release introduces a new property, ReplyCodeActions, for all business process and business operation classes. Formerly, this setting was available only on HL7 TCP business operations. This property allows you to specify how the host should handle each kind of response it receives from the remote system.

This change adds a boolean return value to the existing business operation callback method **OnFailureTimeout()**. If you added an override of this method to your business operation classes, you must add `Quit 0` to your implementation to preserve your custom behavior, and `As %Boolean` to your method signature for it to compile.

This update also changes the format and default behavior of the existing ReplyCodeActions property for HL7 business operations. If you are indicating a literal value found in field MSA:1 or using one of the described special values, you must start your reply code with a colon (:). See the description of the *ReplyCodeActions* property in the `EnLib.HL7.Operation.ReplyStandard` entry in the *Class Reference* for details.

If you upgrade to this release and your production configuration has existing reply codes of this type that do not begin with a colon (:), Ensemble logs warnings in the Event Log for the item by the **OnGetReplyAction()** when the production starts. For example:

```
Unrecognized reply code: '?E'  
Unrecognized reply code: '?R'  
Unrecognized reply code: '~'
```

There were also other changes to the default behavior of properties that may affect your production:

Changes to Default Behavior of HL7 Business Operation Reply Code Actions

A previous release updated and expanded the default behavior of the ReplyCodeActions property with a value of:

```
:?R=RF, :?E=S, :~=S, :?A=C, :*=S, :I?=W, :T?=C
```

This default indicates that Ensemble retries messages with acknowledgment codes AR or CR; for those with codes AE or CE, it suspends the current message, logs an error, and moves on to the next message. This behavior is more consistent with common HL7 processing. The new default also treats any message with codes AA or CA as *Completed OK* and suspends messages that have a value in field MSA:1 that is not matched by any other listed reply code.

Changes to Default Behavior of Business Operation Retry Count

This release redefined the meaning of the `RetryCount` property from “the number of the current try not counting the first try” to “the number of the current try” by setting the default in the business operation class to a value of 1.

13.2.3 New Mechanism for Editing Messages Replaces the `%DrawEditForm()` Method

This release removes the `Ens.Util.MessageBodyMethods.%DrawEditForm()` method, which the Ensemble Management Portal called to display a message-specific content editor. A different mechanism now provides this functionality. See the following sections in the chapter “Viewing, Searching, and Managing Messages” of *Monitoring Ensemble Productions* for details:

- [Resending Messages](#)
- [Managing Suspended Messages](#)

13.2.4 Increased Alert Level for Data Transformation Errors

In previous releases, Ensemble did not trigger an alert when it encountered an error in a data transformation; errors were only logged in the Event Log. Ensemble now reports such errors as alerts if you enable the **Alert On Error** setting for the routing engine configuration item.

13.2.5 Changes to Pool Size Configuration Behavior on TCP Service

For TCP services, when **Job Per Connection** is True, a freshly spawned job handles each new incoming socket connection rather than the listener job itself. Only one job at a time can be the listener, and one job must be the listener, so a TCP service configured with a **Pool Size** greater than 1 still only starts one listener job. However, this listener can spawn an unlimited number of connection jobs if **Job Per Connection** is set to True. If you set the **Pool Size** to a value greater than 1, it serves as a limit on the number of simultaneous connection jobs that can exist. When this limit is reached, the listener does not accept any more connections until one or more of the existing connection jobs quits or dies. An Event Log warning appears when it first reaches the limit.

13.2.6 Renamed Column in Statistics Queries

This release of Ensemble renames a column in the `EnumerateHostStatus()` and `EnumerateJobStatus()` queries in the `Ens.Util.Statistics` class from *LastAction* to *LastActivity*. If your application refers to the column by name, you must update it.

13.2.7 Alert Support for Services Invoked Outside Ensemble

This Ensemble release adds error alerting and logging to the `ProcessInput()` method of business service classes when you invoke the service from a job not started by Ensemble and, therefore, not running in its `OnTask()` loop. The main examples of services invoked in this way are SOAP services and CSP web pages, but may also include language binding and stored procedure calls.

13.2.8 Changes in Empty Schema Category Behavior

In previous releases, if a data transformation processed an HL7 message that had no schema category associated with it, Ensemble modified the source message to have the schema category expected by the data transformation. In this release, the schema category remains empty. It is possible that if a message passed through multiple data transformations or routing engines, it may now fail in subsequent transformations or routing engines. To avoid this problem, specify the schema category in the business service.

14

Ensemble 2008.2

14.1 New Features in Ensemble 2008.2

Ensemble 2008.2 introduced the following new features:

- Digital Signatures and WS-Security
- Ensemble Recovery and Auto-Start
- Caché 2008.2 Features

Ensemble 2008.2 provided enhancements to the following features:

- Licenses and Jobs
- BPL <xpath> Element
- Production-Wide Settings
- Support for Rule Notification
- Default Security Settings for Ensemble CSP Applications
- Revised Error Logging and Handling
- Refinements to ACK and NACK Message Handling
- HTTP Options on HL7 Wizard Pages
- Forced Shutdown Option
- Abort All on Queue Contents Page
- Ensemble Monitoring Using SNMP
- Archive Manager Improvements
- Lookup Table Improvements
- Sequence Manager Improvements

14.2 Compatibility Issues for Upgrades to Ensemble 2008.2

Review the *Caché Release Note and Upgrade Checklist Archive*.

15

Ensemble 2008.1

15.1 New Features in Ensemble 2008.1

Ensemble 2008.1 introduced the following new features:

- MultiValue
- ASTM E 1394–97 Support
- Caché 2008.1 Features

Ensemble 2008.1 provided enhancements to the following features:

- HL7 Segment Architecture Changes
- DTL <subtransform> Element
- Publish and Subscribe Messaging
- HL7 Sequence Manager
- Ensemble Archive Manager
- Ensemble Lookup Settings
- Ensemble System Monitor
- HL7 Version 2 Message Routing
- X12 Message Routing
- Ensemble Monitoring Using WMI
- Ensemble Diagnostic Report
- Ensemble Management Portal Style
- Time Stamp Specifiers

15.2 Compatibility Issues for Upgrades to Ensemble 2008.1

The following changes in the 2008.1 release may affect the operation of your existing system. Review the following issues before upgrading a previous instance of Ensemble:

- [DTL Validation Errors](#)
- [AllowSessions Setting Removed from EnsLib.SOAP.Service](#)

Also review the *Caché Release Note and Upgrade Checklist Archive*.

15.2.1 DTL Validation Errors

In Ensemble 2008.1 and later, including this release, DTL validation is more strict than in the past. As a result, if a DTL code block contains an `<assign>` element with `value= ''` and any of the following action values:

```
action='append'  
action='insert'  
action='set'
```

The code fails to compile, because a non-empty value is mandatory in these cases. Upon upgrade from a previous version to Ensemble 2008.1 and later, errors appear when user classes are recompiled. The error message is:

```
ERROR <Ens>ErrDTLNodeValidation: 'value' must NOT be empty string for action 'Assign'
```

If you have any DTL `<assign>` elements with `value= ''` you must change this text to:

```
value='''
```

This convention adds a pair of double quotes to indicate the null string.

15.2.2 AllowSessions Setting Removed from EnsLib.SOAP.Service

In the 2008.1 Ensemble release, the `AllowSessions` setting was removed from the `EnsLib.SOAP.Service` class. It is no longer configurable; instead, you must choose whether the service should use CSP/SOAP sessions at compile time using the `SOAPSESSION` class parameter. The default for the parameter is now `SOAPSESSION = 0`.

If your subclass of `EnsLib.SOAP.Service` relies on the `AllowSessions` setting to control session behavior, you must rewrite it to use the `SOAPSESSION` class parameter. If you are using sessions you must override it to `SOAPSESSION = 1`. If you do not use sessions, do not override the `SOAPSESSION` class parameter; you can rely on the default setting.

See “[Enabling SOAP Sessions](#)” in the chapter “Creating an Ensemble Web Service” in *Creating Web Services and Web Clients with Ensemble*.

16

Older Releases

The following sections outline the older Ensemble releases:

- [Ensemble 2007.1](#)
- [Ensemble 4.0](#)
- [Ensemble 3.1](#)
- [Ensemble 3.0](#)
- [Ensemble 2.1](#)
- [Ensemble 2.0](#)
- [Ensemble 1.0](#)

Note that for these older releases, there were no compatibility issues specific to Ensemble; therefore, you need only review upgrade checklists in the Caché documentation; see [Caché Release Note](#) and [Upgrade Checklist Archive](#).

16.1 Ensemble 2007.1

Ensemble 2007.1 introduced the following new features:

- Underlying Caché 2007.1 Technologies
- BPL Exception and Compensation Handling
- X12 Support
- Adapter SSL/TLS Support

Ensemble 2007.1 provided enhancements to the following features:

- BPL and XSLT
- BPL Syntax Additions
- BPL Connect Elements
- DTL Syntax Additions
- DTL Visual Editor
- Studio Assist in BPL and DTL

- [Utility Functions](#)
- [Time Stamp Specifiers](#)
- [Link to System Management Portal](#)
- [Alerts in the Configuration Diagram](#)
- [Message Filter and Search](#)
- [Message Browser Indices](#)
- [Message Resend](#)
- [HL7 Routing Productions](#)
- [HL7 Routing Rules](#)
- [HL7 Search Tables](#)
- [HL7 Batch Support](#)
- [HL7 Virtual Properties in Routing Rules](#)
- [HL7 Content in Error Messages](#)
- [HL7 Pages Removed](#)
- [Maintenance Purge Page](#)
- [Trace Messages in the Event Log](#)

16.2 Ensemble 4.0

Ensemble 4.0 introduced the following new features:

- [Underlying Caché 5.2 Technologies](#)

Ensemble 4.0 provided enhancements to the following features:

- [Namespaces](#)
- [Default Login](#)
- [Command Line](#)
- [Management Portals](#)
- [BPL Syntax](#)
- [BPL Visual Editor](#)
- [BPL Components](#)
- [Utility Functions](#)
- [DTL Visual Editor](#)
- [Adapters](#)

16.3 Ensemble 3.1

Ensemble 3.1 introduced the following new features:

- HL7 Support

Ensemble 3.1 provided the following major enhancements:

- Message Contents
- How Data is Purged
- Business Process Context
- Business Process Actor Pools
- Business Hosts
- Ensemble Management Portal
- Configuration Page
- Dashboards
- Business Rules
- Java Gateway
- DTL Visual Editor
- DTL Syntax
- BPL Visual Editor
- BPL Syntax
- Adapters
- Workflow

16.4 Ensemble 3.0

Ensemble 3.0, released in November 2004, was the first public release of Ensemble.

Ensemble 3.0 introduced the following new features:

- Business Activity Monitoring
- Workflow Management
- Business Rules
- Java Gateway

Ensemble 3.0 enhanced the following existing features:

- Business Hosts
- Business Process Language
- BPL Visual Editor

- Management Portal

16.5 Ensemble 2.1

Ensemble 2.1 enhanced the following existing features:

- Business Process Language
- BPL Visual Editor
- Management Portal
- Adapter Library

16.6 Ensemble 2.0

Ensemble 2.0 enhanced existing features and introduced the following new features:

- Universal Services Architecture
- Business Hosts
- Messaging Engine
- Adapter Library
- Business Process Language
- Data Transformation Language
- Management Portal
- Message Visual Trace
- Code Generation Wizards
- BPL Visual Editor
- DTL Visual Editor
- Event Log
- Monitoring Service
- Testing Service
- Documentation
- Sample Code

16.7 Ensemble 1.0

Ensemble 1.0 introduced the following features:

- Application Integration

- Data Integration
- Data Abstraction
- Persistence Engine
- Storage Engine
- SQL Gateway
- Studio

