



# Ensemble Release Notes

Version 2012.1  
30 January 2012

*Ensemble Release Notes*

Ensemble Version 2012.1 30 January 2012

Copyright © 2011 InterSystems Corporation

All rights reserved.

This book was assembled and formatted in Adobe Page Description Format (PDF) using tools and information from the following sources: Sun Microsystems, RenderX, Inc., Adobe Systems, and the World Wide Web Consortium at [www.w3c.org](http://www.w3c.org). The primary document development tools were special-purpose XML-processing applications built by InterSystems using Caché and Java.



Caché WEBLINK, Distributed Cache Protocol, M/SQL, M/NET, and M/PACT are registered trademarks of InterSystems Corporation.



InterSystems Jalapeño Technology, Enterprise Cache Protocol, ECP, and InterSystems Zen 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 Customer Support**

Tel: +1 617 621-0700

Fax: +1 617 374-9391

Email: [support@InterSystems.com](mailto:support@InterSystems.com)

# Table of Contents

<b>About This Book</b> .....	<b>1</b>
<b>1 New Features</b> .....	<b>3</b>
1.1 Redesigned User Interface .....	3
1.2 Granular Security in Management Portal .....	4
1.3 New Business Rules .....	4
1.4 Source Control Hooks in Management Portal .....	4
1.5 Record Mapper .....	4
1.6 Record Batch Handling .....	5
1.7 XML Virtual Documents .....	5
1.8 Alert Generation on Monitored Thresholds .....	5
1.9 UDDI API .....	6
1.10 Caché 2012.1 Features .....	6
<b>2 Enhancements</b> .....	<b>7</b>
2.1 DeepSee Style Dashboards .....	7
2.2 ebXML Messaging .....	7
2.3 Sequence Manager .....	7
2.4 Creating a Studio Project from a Production .....	8
2.5 Documentation Updates .....	8
<b>3 Known Issues</b> .....	<b>11</b>
3.1 EnsLib.HL7.Segment GetValueAt() 32-KB Limitation .....	11
3.2 Internet Explorer 9 Restriction .....	11
3.3 Failure to Add Imported Rules Exported from Prior Versions to a Studio Project .....	12
3.4 SFTP Mode Lacks FileSpec Support in FTP Inbound Adapter .....	12
3.5 Message Browser Search With TimeCreated Property .....	12
3.6 Business Rule Export and Import .....	12
3.7 HL7 Schema Errors .....	13
<b>4 Upgrade Compatibility Issues</b> .....	<b>15</b>
4.1 Compatibility Issues for Upgrades to Ensemble (this release) .....	15
4.1.1 New Management Portal User Interface .....	16
4.1.2 Business Rule Conversion .....	16
4.1.3 Changes in Rule Log Structure .....	16
4.1.4 New Dashboard Development Tool .....	17
4.1.5 New Security Model for Management Portal .....	17
4.1.6 Changes to Workflow User Interface .....	17
4.1.7 Updated Selectivity and Extent Size of the Message Warehouse .....	17
4.1.8 Updated Saved Message Searches .....	18
4.1.9 Removed CSPX Files from Distribution .....	19
4.1.10 Updated Search Table Validation .....	19
4.1.11 New DTL Classes Created with REPORTERERRORS Parameter Set to True .....	19
4.1.12 Updated Legal Character Checking in Configuration Names .....	20
4.1.13 Change in Inactivity Timeout Behavior .....	20
4.1.14 Removed Host Monitor from User Interface .....	20
4.1.15 Improved Notification for Stopping a Running Production .....	20
4.1.16 Change in \$\$\$EnsSystemError Behavior .....	21
4.2 Compatibility Issues for Upgrades to Ensemble 2010.2 .....	21

4.2.1 Remove Support for HL7v2 Framing with XML Text .....	21
4.2.2 Update Error Handling on HTTP Outbound Adapter .....	21
4.2.3 Update Error Processing in File Outbound Adapter .....	22
4.2.4 Change Return Status on HTTP Inbound Adapter .....	22
4.2.5 Add Requirement to Subclass Message Bank Production .....	22
4.2.6 Update Disable Behavior of Business Processes .....	22
4.3 Compatibility Issues for Upgrades to Ensemble 2010.1 .....	23
4.3.1 Relocate RemoveItem() Configuration Method .....	23
4.3.2 Add Configuration Setting on TCP Counted Outbound Adapter .....	23
4.3.3 Correct Type Node in HL7 Sequence Manager Global .....	24
4.3.4 Correct Behavior of HL7 Configuration Framing Setting .....	24
4.3.5 Add Support for Legacy FTPS Protocol to FTP Adapters .....	24
4.3.6 Changes in Mapping of Custom Schemas .....	24
4.4 Compatibility Issues for Upgrades to Ensemble 2009.1 .....	25
4.4.1 Changes in HL7 Storage Structure .....	25
4.4.2 New ReplyCodeActions Property in Process and Operation Classes .....	25
4.4.3 New Mechanism for Editing Messages Replaces the %DrawEditForm() Method .....	26
4.4.4 Increased Alert Level for Data Transformation Errors .....	26
4.4.5 Changes to Pool Size Configuration Behavior on TCP Service .....	26
4.4.6 Renamed Column in Statistics Queries .....	26
4.4.7 Alert Support for Services Invoked Outside Ensemble .....	27
4.4.8 Changes in Empty Schema Category Behavior .....	27
4.5 Compatibility Issues for Upgrades to Ensemble 2008.2 .....	27
4.6 Compatibility Issues for Upgrades to Ensemble 2008.1 .....	27
4.6.1 DTL Validation Errors .....	27
4.6.2 AllowSessions Setting Removed from EnsLib.SOAP.Service .....	28
<b>5 Release History .....</b>	<b>29</b>
5.1 Ensemble 2010.2 .....	29
5.2 Ensemble 2010.1 .....	30
5.3 Ensemble 2009.1 .....	30
5.4 Ensemble 2008.2 .....	31
5.5 Ensemble 2008.1 .....	31
5.6 Ensemble 2007.1 .....	32
5.7 Ensemble 4.0 .....	33
5.8 Ensemble 3.1 .....	33
5.9 Ensemble 3.0 .....	34
5.10 Ensemble 2.1 .....	34
5.11 Ensemble 2.0 .....	35
5.12 Ensemble 1.0 .....	35

# About This Book

Welcome and thank you for using Ensemble 2012.1.

This book describes the contents of this release of Ensemble including compatibility issues with previous releases. It contains the following chapters:

- [New Features](#)
- [Enhancements](#)
- [Known Issues](#)
- [Upgrade Compatibility Issues](#)
- [Release History](#)

There is also a detailed [table of contents](#).

InterSystems Ensemble shares many underlying core technologies with InterSystems Caché. This book refers you to books in both documentation sets. The Caché books are particularly important as you first set up the system. After initial configuration, the Ensemble books become your primary source of information. The following books are the best places to start if you are new to Ensemble:

- *[Preparing to Use Ensemble](#)* provides a road map for installing, upgrading, and getting started with Ensemble.
- *[Introducing Ensemble](#)* provides an overview of product features.
- *[Ensemble Best Practices](#)* describes best practices for designing, developing, and maintaining Ensemble productions.

For general information, see *Using InterSystems Documentation*.



# 1

## New Features

Ensemble 2012.1 offers the following new features since Ensemble 2010.2:

- [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](#)

The next chapter, “[Enhancements](#),” describes changes in existing features since Ensemble 2010.2.

### 1.1 Redesigned User Interface

The Ensemble 2012.1 release contains a completely redesigned web-based user interface for configuring and managing Ensemble solutions. The new Management Portal provides a more modern look and feel and uses the latest web technologies to provide the following features:

- More intuitive flow by keeping the task at hand central to the user activities
- More sophisticated wizards to improve productivity when configuring connections
- New Data Transformation Language (DTL) and Business Process Language (BPL) visual editors
- New business rule editor
- Source control hooks for components developed or amended through the Management Portal, including BPL, DTL, record maps, and business rules
- Native granular security for components of the user interface. A number of predefined roles control access to the Ensemble functions in the Management Portal and enable you to customize what you want to allow for each role or the creation of entirely new roles to provide further customizing.

The new implementation also avoids the duplication of every web page in each namespace, simplifying installation and reducing the time required for upgrades where a system has many namespaces.

If you are upgrading, see the [New Management Portal User Interface](#) section for compatibility issues.

See the *Managing Ensemble* guide for an overview of the Management Portal and where to go to read about a specific Ensemble topic.

## 1.2 Granular Security in Management Portal

This release of Ensemble contains additional roles that you can use to control who has access to different parts of the Management Portal and to different functions. While the built-in roles may suit many customers, it is possible to add additional roles, or to customize role requirements to access pages or functions.

If you are upgrading, see the [New Security Model for Management Portal](#) section for compatibility issues.

See the “[Controlling Access to Management Portal Functions](#)” chapter of *Managing Ensemble* for details.

## 1.3 New Business Rules

Previous to this release, business rules and routing rules have been separate and have been implemented as blocks of free standing XML. Both rules are now defined as classes, extending `Ens.Rule.Definition`, with the rule set implemented in an XDATA block.

While this change in itself is transparent, it does bring immediate advantages in areas such as source control, and the common approach to all aspects such as compilation. In addition to this, the rules XML definition has been enhanced to support nested rules and nested conditions along with an enhanced graphical rules editor which exposes these improvements.

For information on how an upgrade affects your current business rule definitions, see the “[Business Rule Conversion](#)” section.

## 1.4 Source Control Hooks in Management Portal

For a long time, Studio has provided hooks to invoke a source control system when accessing classes. With more classes now being modified in the Management Portal (including rules, record maps, data transformations, and business processes), 2012.1 introduces equivalent hooks for the Management Portal. This allows developers working in a team to check these components in and out when modifying components.

See the [Configure Source Control Settings](#) section of the “Configuring Settings for an Ensemble Namespace” chapter of *Configuring Ensemble Productions* for details.

## 1.5 Record Mapper

The most common form of proprietary formats are files with record structures that have either fixed-width fields or delimited fields including nested fields. The new Ensemble Record Mapper tool provides a graphical interface for describing records in files of this type and prebuilt components for reading or writing files with mapped records.

The Record Mapper generates persistent classes that reflect the structure you define in the user interface, and prebuilt business services read the file and generate request messages. A prebuilt business operation writes to the files with the appropriate layout.

See the new book, *Using the Ensemble Record Mapper* for details.

## 1.6 Record Batch Handling

Closely associated with the Record Mapper is the new Record Batch Handling feature. This allows you to treat input records individually or as a unit. When writing records, this allows you to batch records. For example, you can write all the input records from a particular input file to the same output file, or write all the record received in a given period of time to the same output file.

See the “[Batch Processing and Handling](#)” chapter of *Using the Ensemble Record Mapper* for details.

## 1.7 XML Virtual Documents

Ensemble to date has been able to process XML documents in two ways:

- By correlating them with Ensemble classes, which provides a very powerful way to handle XML with the Ensemble DTL graphical editor. However, for complex XML schemas and large XML documents where only a small number of fields are accessed or modified, the equivalent classes can be complex and correlation can be expensive in system resources.
- By treating the XML as a stream and using XPATH or XSLT to access or modify fields in the document. This is powerful but is technically complicated and requires specialist programming skills.

XML Virtual Documents provide a third option. By using an approach similar to the approach used with EDI formats such as X12, HL7, and DICOM, Ensemble allows you to process XML documents as a stream by using the Ensemble graphical data transformation editor and routing rules. This provides a simpler approach than using XPATH or XSLT, but is lighter weight than object correlation.

See the new *Ensemble XML Virtual Document Development Guide* for details.

## 1.8 Alert Generation on Monitored Thresholds

In addition to existing alerts triggered by error conditions, you can now configure thresholds to control the triggering of alerts when queue depths are too long, messages have been on queues too long, or components have been inactive for too long.

You can define the thresholds on individual components and can control them by time of day.

See the “[Setting, Processing, and Monitoring Alerts](#)” chapter of *Monitoring Ensemble Productions*.

## 1.9 UDDI API

The Universal Description, Discovery and Integration (UDDI) specifications define a registry service for web services and for other electronic and nonelectronic services. This release of Ensemble contains the `EnsLib.UDDI` package to provide an API to access a UDDI registry. See the `EnsLib.UDDI.Base` entry in the *Class Reference* for details.

## 1.10 Caché 2012.1 Features

Ensemble 2012.1 runs on top of Caché 2012.1. This means that, in addition to changes in Ensemble between 2010.2 and 2012.1, the new Ensemble release includes a large number of updates in the underlying Caché technologies. There was also a release of Caché 2011.1 which you may need to review also.

To learn about the Caché changes that may relate to your Ensemble environment, begin with the InterSystems online documentation set called *Getting Started with Caché*. The “New and Enhanced Features for Caché 2012.1” and “New and Enhanced Features for Caché 2011.1” chapters of the *Caché Release Notes* contains details of the Caché 2011.1 and Caché 2012.1 features.

If you are upgrading, see the *Caché Upgrade Checklists*.

# 2

## Enhancements

Ensemble 2012.1 offers enhancements to existing features as follows:

- [DeepSee Style Dashboards](#)
- [ebXML Messaging](#)
- [Sequence Manager](#)
- [Creating a Studio Project from a Production](#)
- [Documentation Updates](#)

### 2.1 DeepSee Style Dashboards

Dashboards in Ensemble have been upgraded to use the same ZEN components used by DeepSee. This not only provides an improved graphical appearance, but also extends the options available with dashboards you create for Ensemble data.

For information on how an upgrade affects your current dashboards, see the “[New Dashboard Development Tool](#)” section.

### 2.2 ebXML Messaging

With this release, Ensemble extends its existing support for ebXML messaging in a very specific direction. ebXML is a very large standard; this release of Ensemble implements additional message patterns that apply very specifically to connectivity within the English National Health Service.

### 2.3 Sequence Manager

The sequence manager has been enhanced to allow more than one sequence manager in a production. See the “[HL7 Sequence Manager](#)” section of the “Elements of a Routing Production” chapter of the *Ensemble HL7 Version 2 Development Guide* for details.

## 2.4 Creating a Studio Project from a Production

The capability to create a studio project from a production helps with the export of productions from one system and import into another. This has been enhanced to include HL7 schemas, subclasses of routing engines, and subtransforms referenced in the production as part of the generated project.

For details, see the “[Using Studio in Ensemble](#)” section of *Developing Ensemble Productions*.

## 2.5 Documentation Updates

Ensemble 2012.1 introduces the following new books:

- *Ensemble 2012.1 Release Notes* (this book) — Describes the contents of this release of Ensemble as well as compatibility issues with previous releases.
- *Preparing to Use Ensemble* — Provides a road map for installing, upgrading, and getting started with Ensemble.
- *Using the Ensemble Record Mapper* — Explains how to use the Record Mapper to map data in text files to persistent Ensemble messages.
- *Ensemble XML Virtual Document Development Guide* — Explains how to build Ensemble productions that route and transform documents in XML format.
- *Configuring Ensemble Productions* — Explains how to build and configure Ensemble productions and production components as well as update configurations of existing productions.
- *Monitoring Ensemble* — Explains how to monitor your Ensemble environment and the various Ensemble production components.

Ensemble 2012.1 offers significant updates to the following books:

- *Managing Ensemble*
  - Complete restructuring to be a companion to the new Management Portal user interface.
  - Many sections moved to the *Configuring Ensemble Productions* and *Monitoring Ensemble* books.
  - Corrections and updates to several topics.
- *Developing Ensemble Productions*
  - Updates to the wizard descriptions to reflect additional wizard processing
  - Added chapter on “[Developing BPL Processes](#)” to reflect new user interface in the Management Portal, the Business Process Designer.
  - Added chapter on “[Developing DTL Documents](#)” to reflect new user interface in the Management Portal, the Data Transformation Builder.
  - Minor corrections and updates to several topics.
- *Using Business Rules with Ensemble* to reflect the new user interface in the Management Portal, the Business Rule Editor.
- *Using Dashboards with Ensemble* to reflect the relocation of the creating dashboards feature to the DeepSee portion of Caché. *Creating DeepSee II Dashboards* describes the relevant material.

- [Using Workflow with Ensemble](#) to reflect the relocation of the supervisory workflow tasks to the DeepSee portion of Caché and the new user interface in the Management Portal.

Ensemble 2012.1 contains a renamed book:

- [Managing Ensemble Productions](#) is now named [Managing Ensemble](#).

Ensemble 2012.1 offers user interface and other minor corrections and updates to the following books:

- [Ensemble Business Process Language Reference](#)
- [Ensemble Data Transformation Language Reference](#)
- [Ensemble Virtual Documents](#)
- [Ensemble HL7 Version 2 Development Guide](#)
- [Ensemble HL7 Version 3 Development Guide](#)
- [Ensemble X12 Development Guide](#)
- [Ensemble ASTM Development Guide](#)
- [Ensemble EDIFACT Development Guide](#)
- [Ensemble DICOM Development Guide](#)
- [Using Email Adapters with Ensemble](#)
- [Using File Adapters with Ensemble](#)
- [Using FTP Adapters with Ensemble](#)
- [Using HTTP Adapters with Ensemble](#)
- [Using IBM WebSphere MQ Adapters with Ensemble](#)
- [Creating Web Services and Web Clients with Ensemble](#)
- [Using SQL Adapters with Ensemble](#)
- [Using TCP Adapters with Ensemble](#)



# 3

## Known Issues

Review the following issues carefully to determine if they affect your system:

- “Upgrade Compatibility Issues” described in the next chapter.
- [EnsLib.HL7.Segment GetValueAt\(\) 32-KB Limitation](#)
- [Internet Explorer 9 Restriction](#)
- [Failure to Add Imported Rules Exported from Prior Versions to a Studio Project](#)
- [SFTP Mode Lacks FileSpec Support in FTP Inbound Adapter](#)
- [Message Browser Search With TimeCreated Property](#)
- [Business Rule Export and Import](#)
- [HL7 Schema Errors](#)

**Important:** For an updated list of known problems in this release, contact the [InterSystems Worldwide Response Center \(WRC\)](#).

### 3.1 EnsLib.HL7.Segment GetValueAt() 32-KB Limitation

The **GetValueAt()** method of the EnsLib.HL7.Segment class truncates values larger than 32 KB. This limitation exists even if you have long strings enabled.

As a work-around, use the **ReadRawDataStream()** method and then use the \$Piece function to retrieve the portion of the string that comprises the field you are retrieving.

### 3.2 Internet Explorer 9 Restriction

If you are running Ensemble in Internet Explorer 9, do *not* use the *Compatibility View*.

## 3.3 Failure to Add Imported Rules Exported from Prior Versions to a Studio Project

If you import an XML export containing a business rule or routing rule that you exported from a version prior to 2012.1, the import does not add that rule definition to a project in Studio. You receive an error message indicating that the rules does not exist on the server. This occurs because the process attempts to add the old .RUL form of name to the project, but the business rule has been converted to a class. The class containing the rule is created and you can add the class to your project manually.

## 3.4 SFTP Mode Lacks FileSpec Support in FTP Inbound Adapter

The current implementation of SSH File Transfer Protocol (SFTP) mode on the FTP inbound adapter does not support the FileSpec property.

To use the SFTP mode enter `!SFTP` as the value for the existing **SSL Config** setting. For more information, see the description of the `EnsLib.FTP.InboundAdapter` class in the *InterSystems Class Reference*.

## 3.5 Message Browser Search With TimeCreated Property

When using the message browser after an upgrade and specifying a **Start Time**, exact matches against the start time are not shown in some circumstances. If the time you enter would end in one or more trailing zeroes when the seconds are expressed to three decimal places and there is a message created before upgrading at that exact time, that message is not included in the search result.

For example, Ensemble creates a message prior to the upgrade at `2009-12-02 15:16:44.710`. After the upgrade, if you enter `2009-12-02 15:16:44.710` or `2009-12-02 15:16:44.71` as the **Start Time** in your search criteria, the message is not found. To work around this issue, widen the search time slightly to `2009-12-02 15:16:44.709`.

You could resolve the problem by rebuilding the `TimeCreated` index of the `Ens.MessageHeader` class, but InterSystems does not recommend this for most customers. It requires the system to be idle during the rebuild, which can take several hours for message warehouses with 100 million messages. Since most searches are for recent messages, this is expected to only present a problem for a short period after upgrading. Similar behavior exists when using SQL searches against the `Ens.MessageHeader` class. This issue also exists in Ensemble release 2009.1.

## 3.6 Business Rule Export and Import

InterSystems has identified a known problem with the Xerces parser version used in the current and past releases for Ensemble. The symptom related to Ensemble business rules is that Ensemble wrongly reports errors when importing a previously exported production from an XML file. The symptom occurs only when the XML file contains definitions of general business rules that define “assign” actions in addition to simply returning a result.

There are two techniques for working around this problem. One makes import simple and places the burden on the person exporting the production. The other makes export simple and places the burden on the person importing the production. You only need to use one of the following equally effective techniques:

### Import

Use the following approach to facilitate the import task:

1. Find each general business rule that defines “assign” actions in addition to returning a result.
2. Export each of these rules to a separate file. Make sure you are exporting one rule per file.
3. Export everything else in the production, including other rules, to a different file.
4. Import (and compile) each of the exported files individually.

### Export

Use the following approach to facilitate the export task:

1. Export everything to one file.
2. Upon Import, do not use Studio. Instead, start Terminal, change to the namespace where you need to import, and enter one of the following commands (either works):

```
do $system.OBJ.Load("C:\MyDir\MyFile.xml", "-i")  
  
do $system.OBJ.Load("C:\MyDir\MyFile.xml", "/checkschema=0")
```

## 3.7 HL7 Schema Errors

The HL7 schema definitions loaded into Ensemble were generated directly from the respective standards (HL7 2.1, 2.2, 2.3, 2.3.1, 2.4, 2.5, 2.5.1, and 2.6). They replicate any errors, omissions, or discrepancies that exist in these standards as published by the Health Level Seven organization. There are a few known exceptions, as follows:

- In the HL7 2.3.1 standard, the data structure XCN is the “extended composite ID number and name for persons.” The standard leaves XCN field 3 undefined by mistake. The Ensemble schema definition for HL7 2.3.1 corrects this so that XCN field 3 is correctly identified as “given name.”
- In the HL7 2.5, 2.5.1, and 2.6 standards, the segment structure SUR\_P09 refers to undefined segment type ED. The Ensemble schema definitions correct this by renaming the reference to type ED1 and defining a segment type ED1 that contains one field named “Data” with data structure type ED.
- In the HL7 2.3 standard, the ADT\_A37 message type contains the incorrect message structure of ADT\_A37 which does not exist in this HL7 version. The Ensemble schema definition for HL7 2.3 corrects the message structure to ADT\_A24. In later HL7 versions there is a message structure ADT\_A37 so no correction is necessary.
- In the HL7 2.4 standard, the OBR segment structure contains mislabeled data structures in Field 15; namely there are CE data structures that have their code table number appended to them (CE0070, for example). The Ensemble schema definition for HL7 2.4 changes the data structures named CE0070, CE0163, and CE0369 to CE.



# 4

## Upgrade Compatibility Issues

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:

- [Compatibility Issues for Upgrades to Ensemble \(this release\)](#)
- [Compatibility Issues for Upgrades to Ensemble 2010.2](#)
- [Compatibility Issues for Upgrades to Ensemble 2010.1](#)
- [Compatibility Issues for Upgrades to Ensemble 2009.1](#)
- [Compatibility Issues for Upgrades to Ensemble 2008.2](#)
- [Compatibility Issues for Upgrades to Ensemble 2008.1](#)

The following releases did not include compatibility issues specific to Ensemble; therefore, you need only review the Caché documentation:

- *Caché 2007.1 Upgrade Checklist*
- *Caché 5.2 Upgrade Checklist*

### 4.1 Compatibility Issues for Upgrades to Ensemble (this release)

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 2011.1 Upgrade Checklist* and the *Caché 2012.1 Upgrade Checklist*.

## 4.1.1 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.

## 4.1.2 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.

## 4.1.3 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.

## 4.1.4 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 “[Adding Business Metrics to Dashboards](#)” chapter of *Using Dashboards with Ensemble* for guidance on where to begin.

## 4.1.5 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, so that they can perform the same tasks as before. 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 the “[Controlling Access to Management Portal Functions](#)” chapter of *Managing Ensemble*.

## 4.1.6 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.

For details, see [Using Workflow with Ensemble](#).

## 4.1.7 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. Then on the **View** menu, click **View Storage** to see the storage definition displayed at the end of the class definition. It includes `<ExtentSize>` and `<Selectivity>` entries in a form similar to the one previously listed for this release.
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 the “ExtentSize and Selectivity” section of the “Optimizing Performance” chapter of *Using Caché SQL*.

## 4.1.8 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 2012.1 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()
```

## 4.1.9 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 made use of these files, contact the [InterSystems WRC](#) for help in upgrading your Ensemble environment.

## 4.1.10 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.

## 4.1.11 New DTL Classes Created with REPORTERRORS Parameter Set to True

Previous to this release, all DTL data transformation classes inherited from `Ens.DataTransformDTL` had the `REPORTERRORS` 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. See the “[Using the Data Transformation Wizard](#)” section in the “Developing DTL Transformations” chapter of *Developing Ensemble Productions* for more information.

## 4.1.12 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.

## 4.1.13 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.

## 4.1.14 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.

## 4.1.15 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.

## 4.1.16 Change in \$\$\$EnsSystemError Behavior

In previous releases, the \$\$\$EnsSystemError macro logged all exceptions it trapped in Ensemble to the %ETN routine. 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 non-zero 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 made use of 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.

## 4.2 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 2010.2 Upgrade Checklist*.

### 4.2.1 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](#).

### 4.2.2 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 the [Reply Code Actions Setting](#) section of *Configuring Ensemble Productions* for details.

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 the [HL7 ACK Configuration Settings for a Business Operation](#) section of the “Settings for a Routing Production” chapter of *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.

### 4.2.3 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`.

### 4.2.4 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.

### 4.2.5 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.

### 4.2.6 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 the “[Pool Size](#)” section in the “[Production Concepts](#)” chapter of *Developing Ensemble Productions*.

## 4.3 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 2010.1 Upgrade Checklist*.

### 4.3.1 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.

### 4.3.2 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.

### 4.3.3 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.

### 4.3.4 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.

### 4.3.5 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.

### 4.3.6 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.

## 4.4 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 2009.1 Upgrade Checklist*.

### 4.4.1 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.

### 4.4.2 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 `EnsLib.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.

## 4.4.3 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 “Viewing, Searching, and Managing Messages” chapter of *Monitoring Ensemble Productions* for details:

- [Resend Messages](#) section of the “Message Browser” chapter
- [Resubmitting Suspended Messages](#)

## 4.4.4 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.

## 4.4.5 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.

## 4.4.6 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.

## 4.4.7 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.

## 4.4.8 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.

# 4.5 Compatibility Issues for Upgrades to Ensemble 2008.2

Review the *Caché 2008.2 Upgrade Checklist*.

## 4.6 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 2008.1 Upgrade Checklist*.

### 4.6.1 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.

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

# 5

## Release History

The Ensemble product was developed to meet the need for a comprehensive, high-performance enterprise application and data integration platform with tightly integrated development, management, and supervisory tools.

The following sections outline a brief history of Ensemble releases starting with the most recent:

- [Ensemble 2010.2](#)
- [Ensemble 2010.1](#)
- [Ensemble 2009.1](#)
- [Ensemble 2008.2](#)
- [Ensemble 2008.1](#)
- [Ensemble 2007.1](#)
- [Ensemble 4.0](#)
- [Ensemble 3.1](#)
- [Ensemble 3.0](#)
- [Ensemble 2.1](#)
- [Ensemble 2.0](#)
- [Ensemble 1.0](#)

### 5.1 Ensemble 2010.2

Ensemble 2010.2 was released in October 2010 and introduced the following new features:

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

Ensemble 2010.2 offered 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

Also see the *New and Enhanced Features for Caché 2010.2* in the *Caché Release Notes*.

## 5.2 Ensemble 2010.1

Ensemble 2010.1 was released in February 2010 and introduced the following new features:

- Configuration Default Settings
- EDIFACT Support
- Caché 2010.1 Features

Also see the *New and Enhanced Features for Caché 2010.1* in the *Caché Release Notes*.

## 5.3 Ensemble 2009.1

Ensemble 2009.1 was released in July 2009 and 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 offered 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

Also see the *New and Enhanced Features for Caché 2009.1* in the *Caché Release Notes*.

## 5.4 Ensemble 2008.2

Ensemble 2008.2 was released in October 2008 and introduced the following new features:

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

Ensemble 2008.2 offered 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

Also see the *New and Enhanced Features for Caché 2008.2* in the *Caché Release Notes*.

## 5.5 Ensemble 2008.1

Ensemble 2008.1 was released in July 2008 and introduced the following new features:

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

Ensemble 2008.1 offered 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

Also see the *New and Enhanced Features for Caché 2008.1* in the *Caché Release Notes*.

## 5.6 Ensemble 2007.1

Ensemble 2007.1 was released in October 2007 and introduced the following new features:

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

Ensemble 2007.1 offered 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

Also see the *New and Enhanced Features for Caché 2007.1* in the *Caché Release Notes*.

## 5.7 Ensemble 4.0

Ensemble 4.0 was released in June 2006 and introduced the following new features:

- Underlying Caché 5.2 Technologies

Ensemble 4.0 offered 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

Also see the *New and Enhanced Features for Caché 5.2* in the *Caché Release Notes*.

## 5.8 Ensemble 3.1

Ensemble 3.1 was released in April 2006 and introduced the following new features:

- HL7 Support

Ensemble 3.1 offered 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

## 5.9 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

## 5.10 Ensemble 2.1

Ensemble 2.1 was released to InterSystems customers in November 2003; it enhanced the following existing features:

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

## 5.11 Ensemble 2.0

Ensemble 2.0 was released to InterSystems customers in August 2003. This release 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

## 5.12 Ensemble 1.0

Ensemble 1.0 was released to InterSystems customers in 2002 and introduced the following features:

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

