InterSystems® Ensemble

Managing Ensemble

Version 2018.1
2019-09-20

Managing Ensemble
Ensemble   Version 2018.1   2019-09-20
Copyright © 2019 InterSystems Corporation
All rights reserved.

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

This book, which is primarily for system administrators, explains how to use the Management Portal to manage Ensemble productions. It contains the following sections

- Introduction to Ensemble Management

- Starting and Stopping Productions

- Purging Production Data

- Using the Archive Manager

- Managing Workflow Roles, Users, and Tasks

- Defining Publish and Subscribe Message Routing

- Controlling Ensemble Data Storage

- Controlling Access to Management Portal Functions

- Finding Information on Ensemble Menu Items

For a detailed outline, see the table of contents.

The following books are also relevant to system administrators:

- *Configuring Ensemble Productions* describes how to configure the settings for Ensemble productions, business hosts, and adapters.

- *Monitoring Ensemble* explains how to monitor your Ensemble environment and the production components.

- *Caché System Administration Guide* describes common system administration tasks in a pure Caché environment.

- *Caché Security Administration Guide* describes authentication, authorization, auditing, managed key encryption, SSL/TLS, and other aspects of Caché security.

For general information, see the *InterSystems Documentation Guide*.

1

Introduction to Ensemble Management

This chapter introduces the Management Portal and the tasks involved in managing Ensemble. It includes the following topics:

- Background for System Administrators
- Introduction to Ensemble Management
- Introduction to the Management Portal
- Getting Started with the Management Portal
- Viewing Summaries for Active Productions

This book provides information on options related to managing Ensemble; this is a subset of the options in the **Ensemble** menu. To find information on other options in the **Ensemble** menu, see the appendix "Finding Information on Ensemble Menu Items."

For information on the other menus of the Management Portal, see the *Caché System Administration Guide*.

1.1 Background for System Administrators

This section provides some basic terminology to help you get started.

An Ensemble *production* is a specialized package of software and documentation that integrates multiple, potentially disparate software systems. A production includes elements that communicate with these external systems, as well as elements that perform processing that is internal to the production.

A production consists of a number of *business hosts* that communicate with each other (and with external systems). There are three distinct types of business host:

- A *business service* receives input from outside the production.
- A *business process* is responsible for communication and logic that is entirely within the production.
- A *business operation* usually sends output from the production. Business operations can also be used for communication and logic within a given production.

Within a production, all communication is carried out by means of request and response messages between the business hosts.

Ensemble permits only one production to be running in a given namespace at any given time.

A running production continues to run even when you close the Management Portal.

For additional background, see "Core Concepts" in *Monitoring Ensemble*.

1.2 Introduction to Ensemble Management

The process of managing Ensemble includes the following tasks, all described in this book.

- Purging old data that is no longer needed.

  Ensemble stores messages, Event Log entries, business rule log entries, and other historical data. It is generally necessary to purge old data periodically.

- Enabling or disabling the auto-start option for productions. This option causes a production to start automatically when Ensemble starts, and to shut down when Ensemble is stopped.

- Creating and maintaining workflow roles and users, if any productions use the Ensemble workflow engine.

  It is also possible for supervisors (with sufficient permissions) to assign or cancel workflow tasks.

- Using the Archive Manager, which can archive messages to a separate archive. A newer, preferred option is to use the Enterprise Message Bank, which enables you to archive messages from multiple productions. See "Using the Enterprise Message Bank" in *Monitoring Ensemble*.

- Defining publish and subscribe message delivery.

For information on monitoring the productions — viewing message queues, viewing the Event Log, and examining other such data, see *Monitoring Ensemble*.

1.3 Introduction to the Management Portal

The Management Portal includes both Caché and Ensemble options. Ensemble is built on Caché, so all the Caché options apply equally well to Ensemble. The **Ensemble** menu provides options that apply specifically to Ensemble.

Developers use the portal to configure and deploy new productions. System administrators use the portal to monitor or configure productions that are already running. Business analysts use the portal to define business rules for existing productions.

Note: You use the portal to start a production. However, if a production is running, it continues to run even if the portal is closed. That is, you can safely exit the portal and close the browser; these actions do not affect the current state of any productions.

1.3.1 Browser Requirements

The Management Portal runs on all InterSystems supported web browsers. See "Supported Web Browsers" in the online *InterSystems Supported Platforms* document for this release for details.

On Internet Explorer 9, do not use the *Compatibility View*.

1.4 Getting Started with the Management Portal

1.4.1 Accessing the Management Portal

To access the Management Portal, do any of the following:

- Click the InterSystems launcher ▤ in the Windows system tray and click **Management Portal**.

- In Studio, click **Utilities**, and then click **Management Portal**.

- Use a previously saved bookmark to this page.

1.4.2 Choosing a Namespace

The title bar of the Management Portal provides a **Switch** command that you can click to switch to a different Ensemble namespace.

When you choose the **Ensemble** menu, if you have not selected an Ensemble-enabled namespace, you are prompted to switch to a different namespace:

Click an Ensemble-enabled namespace. For information about namespaces in Ensemble, see "Environmental Considerations" in *Developing Ensemble Productions*.

1.5 Viewing Summaries for Active Productions

When you click any option in the **Ensemble** menu, the right side of the page displays summary information about the productions, as follows:
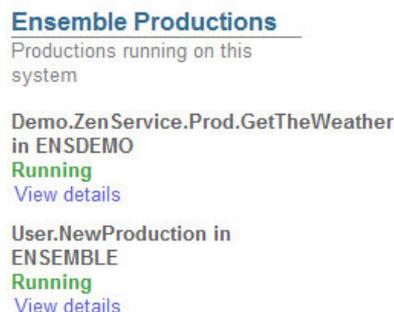
If a production is Suspended or Troubled, see "Correcting Production Problem States" in *Monitoring Ensemble*.

2

Starting and Stopping Productions

By default, Ensemble does not automatically start productions. This chapter describes how to start and stop productions. It discusses the following topics:

- Starting a Production

- Stopping a Production

- Managing Production Auto-Start

- Overriding the Production Auto-Start Options in All Namespaces

- Redeploying a Live Production

Note: For a live, deployed production, InterSystems recommends that you use the auto-start option. The other options are intended primarily for use during development.

2.1 Starting a Production

To start an Ensemble production from the Management Portal:

1. Click **Ensemble** > **List** > **Productions**.

   Ensemble then displays the **Production List** page.

2. Click the production that you want to start.

3. Click Open.

   Ensemble displays the production.

4. Click **Start**.

5. Click **OK**. Ensemble displays a dialog box that indicates progress.

   Note: If any Terminal windows open as a result of starting the production, do not close them.

6. When the dialog box shows it is Done, click **OK**.

2.2 Stopping a Production

To stop an Ensemble production from the Management Portal:

1. Click **Ensemble** > **List** > **Productions**.

    Ensemble then displays the **Production List** page.

2. Click the production that you want to stop. This must be a production that is running.

3. Click Open.

    Ensemble displays the production.

4. Click **Stop**.

5. Click **OK**. Ensemble displays a dialog box that indicates progress.

    Note: If any Terminal windows open as a result of starting the production, do not close them.

6. When the dialog box shows it is Done, click **OK**.

7. If the request to stop the production initially fails, the portal displays a message:

    "Production could not stop, do you want to force a shut down?"

    And provides a command:

    Yes - Force to Shut Down

    If you click this command, the production is forced to shut down.

If a production is Suspended or Troubled, see "Correcting Production Problem States," in *Monitoring Ensemble*.

2.3 Managing Production Auto-Start

You can specify that a production is automatically started in a namespace at system startup, and is automatically stopped at system shutdown. This option is the recommended way to start and stop productions.

To access this page in the Management Portal, select **Ensemble**, click **Manage**, and then click **Auto-Start Production**.

To enable auto-start for a single production in the current namespace:

1. Select the production from the drop-down list.

2. Click **Apply**.

Ensemble displays a dialog asking you to confirm that you want to auto-start this production.

To disable auto-start in the current namespace:

1. Do not select any productions from the drop-down list.

2. Click **Apply**.

Ensemble displays a dialog box asking you to confirm that you do not want to auto-start any production in this namespace.

A different page lets you override auto-start for all productions.

Important: If you define a production to auto-start in a mirror configuration, it automatically starts on the current primary node in a failover situation. No further action is necessary. For details on the mirror failover process, see "Mirroring" in the *Caché High Availability Guide*.

2.4 Overriding the Production Auto-Start Options in All Namespaces

For debugging purposes in a development environment, you can override the auto-start options for all productions. To do so:

1. In the Management Portal, click **System Administration** > **Configuration** > **Additional Settings** > **Startup Settings**.

   This displays the **Startup Settings** page.

2. Click **Edit** next to the **EnsembleAutoStart** setting.

3. Clear the check box.

4. Click **Save**.

When you next restart Ensemble, Ensemble does not automatically start any productions in any namespace (except for any productions that you automatically start in some way other than the auto-start settings documented in this chapter).

2.5 Redeploying a Live Production

The Management Portal automates the process of deploying a production from a development system to a live system. Overview of Deploying a Production describes this process from the developer's point of view. This section describes what Ensemble does when you are loading a new version of a production on a live system.

The developer has provided you with an XML deployment package file that contains an updated version of your production. This deployment package should be deployed to a test system before deploying it to the live system. To load it on the live system, select the correct namespace and select **Ensemble**, **Manage**, **Deployment Changes**, **Deploy**, and then click the **Open Deployment** button and select the file containing the XML deployment package. The form lists the new and changed items in the deployment package, displays the deployment notes that were specified when the package was created.

You can specify the following deployment settings:

- Target production—specifies the production that the components will be added to. If the deployment package includes the production class from the source production, then the target production is set to the source production and cannot be changed. Otherwise, Ensemble sets the default production to the currently open production, but allows you to change it.

- Rollback file—specifies the file to contain the rollback information.

- Deployment log file—contains a log of the changes caused by the deployment.

When you have read the deployment notes and made any changes to the deployment settings, complete the deployment by clicking the **Deploy** button. Ensemble does the following to stop the production, load the new code, and then restart the production.

1. Create and save the rollback package.

2. Disable the components in the production that have a production settings (ptd) file in the deployment package.

3. Import the XML file and compile the code. If there is an error compiling any component, the entire deployment is rolled back.

4. Update the production settings.

5. Write a log detailing the deployment.

6. Enable the production components that were disabled if their current setting specify that they are enabled.

To undo the results of this deployment change, you can select the rollback file with an **Open Deployment** and then click the **Deploy** button.

If you are using studio, you use the **Tools Import Local** or **Import Remote** to import the XML file. But you must then manually compile the components and then disable and re-enable them in the production. Using the **Import** classes button on the Management Portal **System Explorer** does automatically compile the classes, but it does not create a rollback package and disable the components.

2.6 Displaying the Deployment History

You can view the deployment history of the productions in a namespace. To view the deployment history, select **Ensemble**, **Manage**, **Deployment Changes**, and **History**.

After you select one of the listed deployments, you can click **Details** to display information about the deployment, **Rollback** to undo the deployment changes, or **Delete** to delete the deployment history. Deleting the deployment history does not delete the rollback or log files.

3

Purging Production Data

The **Purge Management Data** page allows you to delete outdated entries from the Event Log, message warehouse, business process log, business rule log, and I/O archive log. You can only purge entries associated with the active namespace. To navigate to this page, click **Ensemble** > **Manage** > **Purge Management Data**.

This chapter contains the following topics:

- First-time Purges

- Purging Data Manually

- Purging Data Periodically

3.1 First-time Purges

The activity of purging generates extra journaling, especially if you purge a large volume of data, and this journaling can consume a large amount of disk space. For performance reasons, you might adopt the following approach when you first purge management data:

1. Start by setting the purge parameters so that a relatively small amount of data is purged.

   As discussed in the next section you use the option **Do not purge most recent** to specify how many days' worth of records to keep. Set this to a relatively large number.

2. Perform the purge.

3. Gradually decrease the number of days of data to keep until **Do not purge most recent** is set as desired.

3.2 Purging Data Manually

Typically you purge data manually during development and testing. (In contrast, for a live system, you typically purge data on a periodic basis, as described in the next section.)

To purge data manually, use the **Purge Management Data** page. This page displays the following information:

- **Type of Record** — Identifies the purpose of each row in the table. Each row contains one type of artifact that running productions produce on an ongoing basis: **Event Log**, **Messages**, **Business Processes**, **Business Rule Log**, **I/O Log**, or **Managed Alerts**.

- **Count** — Total number of entries of this type stored for this production. Use the **Count** to decide whether or not it is worthwhile to purge the records and if so, how many days' worth of records you want to keep.

- **Delete** — Total number of entries of this type that the purge process deleted after you click **Start Purge** and the purge completes.

Purge Criteria

Beneath the table is a box to enter your purge criteria and the command to perform the purge.

- **Include message bodies** — If selected, this check box indicates that when Ensemble purges message headers, it should also purge the associated message bodies. Ensemble verifies that body classes exist and are persistent, before purging them.

  If this check box is clear (the default), message header data is purged, but message body data is retained. If you purge message headers but retain the message bodies, then the Management Portal provides no way to purge the orphaned message bodies. In this case, you must purge the message bodies programmatically. In the ENSDEMO database, the class Demo.Util.CleanupSet provides an example of how you might do this.

  Important: When Ensemble purges a message body, it does not necessarily delete object-valued properties of the message body. Specifically, it deletes other objects only if these are serial objects or are child objects (as defined by a relationship). For other objects, you must appropriately handle the deletion by defining a delete trigger or implementing the **%OnDelete()** method in the message body class.

- **Keep data integrity** — If selected (the default) this check box indicates that when Ensemble purges message headers, even if a message meets the age criterion for purging, it is not deleted unless its status is *complete*. Ensemble considers messages to be *complete* if they are marked Complete, Error, Aborted, or Discarded. This is to keep session-level integrity.

  The query that identifies the messages to delete checks all the messages (including business process instances) in a session to see whether any of them are not complete. The purge only performs the delete if all the messages are complete. The scope of this query, therefore, has an impact on the time taken to do the purge.

  This option is important to support long-running business processes. Usually this is the desired behavior. However, if you know there are old messages in the system whose incomplete status is not significant, you can purge them by clearing the **Keep data integrity** check box.

  Important: The purge criteria for using **Keep data integrity** also includes Ensemble system processes such as the Scheduler. Before clearing this check box, carefully consider the value for the **Do not purge most recent** setting.

- **Do not purge most recent** — This tells Ensemble how many days' worth of records to keep. The number can be 0 (zero), which keeps nothing and deletes all entries that exist at the time of the purge operation. The default value for **Do not purge most recent** is 7, which keeps entries for the last seven days.

  The count of days includes *today*, so keeping messages for 1 day keeps the messages generated on the current day, according to local server time.

At the bottom of the *Purge Criteria* box is the **Start Purge** command. If you click it, Ensemble immediately purges the persistent store according to the parameters you have entered. The page uses a background job to do purges, and reports the results of the last-run purge, including a status code, or a notice if the background job is running or has failed to run. When the purge completes, the **Deleted** column contains the number of records purged for each type. If a purge is currently executing in the namespace, the **Start Purge** command is inactive.

CAUTION: You cannot undo the **Start Purge** operation.

3.3 Purging Data Periodically

For a live system, you typically purge data on a periodic basis. To do so

1. Click **System Operation**, click **Task Manager**, and then click **New Task**.

2. For **Task name**, enter a task name.

3. For **Namespace to run task in**, select the applicable namespace.

4. For **Task type**, select Ens.Util.Tasks.Purge.

   When you select this class, the page updates to display configurable information for this task.

5. Specify the following options:

   - **BodiesToo** — Corresponds to the **Include message bodies** option. It is very important to purge message bodies; consequently, this option should be set to true in most cases. If you are not regularly purging message bodies, you should provide another mechanism to ensure that this storage is freed.

   - **KeepIntegrity** — Corresponds to the **Keep data integrity** option.

   - **NumberOfDaysToKeep** — Corresponds to the **Do not purge most recent** option.

   - **TypesToPurge** — Select the type of data to purge.

6. Specify other options as needed. For details, see "Using the Task Manager" in the chapter "Managing Caché" in the *Caché System Administration Guide*.

7. Complete the wizard as described in the previously referenced section.

4

Using the Archive Manager

The **Archive Manager** page allows you to periodically save older messages to a separate archive for long-term storage. The Archive Manager is deprecated. Instead of using the Archive Manager, you should use the Enterprise Message Bank, which enables you to archive messages from *multiple* productions. For an overview, see "Defining the Enterprise Message Bank" in *Developing Ensemble Productions*. Also see "Using the Enterprise Message Bank" in *Monitoring Ensemble*

To navigate to the **Archive Manager** page, click **Ensemble** > **Manage** > **Local Archive Manager**.

This chapter describes how to use this page. It contains the following topics:

- Archive Basics
- Archiving Data
- Default Behavior

4.1 Archive Basics

The page displays the following archive settings for the active namespace:

- **Archive to namespace** — The namespace to which Ensemble saves archived messages.

- **Archive manager class name** — The class that acts as the Archive Manager. Use Ens.Archive.Manager or a custom class, if available.

    See "Defining a Custom Archive Manager" in *Developing Ensemble Productions*.

- **Number of days before archiving** — Messages older than this number of days are automatically archived when you run the archive operation.

    If the Archive Manager performs purging, note that the activity of purging generates extra journaling, especially when you purge large volumes of data; see "First-time Purges," earlier in this book.

The **Archive Manager** requires you to identify a namespace in which to keep the archive. InterSystems strongly recommends that you keep this archive in a namespace that meets *both* of the following criteria:

- A separate namespace from those in which you run Ensemble productions. If you are running productions in more than one namespace, be aware that multiple namespaces can archive their messages into one shared target namespace.

- An Ensemble-enabled namespace, so that you can use the Management Portal features such as the Message Viewer and Visual Trace whenever you have a reason to examine the archived messages. For details about Ensemble-enabled namespaces, see "Environmental Considerations" in *Developing Ensemble Productions*.

Click **Edit** to the right of the namespace to update these settings. Change the information in the fields and then click **Save**. If the save is successful, the page is refreshed with the new settings displayed. If the save failed, the form displays the error message from the server.

The **Archive history** display provides information about the last or current archive. For example:

```
Archive start time  2012-01-05 12:06:10
Archive stop time  2012-01-05 12:06:10
Total messages processed  70 - 100% finished
Total messages archived  0
Total message headers deleted  0
Total message bodies deleted  0
Archive status  idle
```

4.2 Archiving Data

Note: During this process, Ensemble scans the entire message header table. Accordingly, you should use this option when the performance impact is acceptable.

At the bottom of the page is the **Run Archive** command. This command is operable only if there is data in all three fields in the form and there is no previous archive operation still in progress. After clicking **Run Archive**, click **OK** to verify and begin the archive.

CAUTION: You cannot stop the archive operation.

The archive operation runs in the background and displays progress statistics while it is running. The numbers in the display update continuously, with count and percentage continuing to change until the result reaches 100%, status becomes idle, and a final stop time appears:

```
Archive start time: 2008-05-14 18:19:02
Archive stop time:
Total messages processed 100 - 10% finished
Total messages archived 3
Total message headers deleted 1
Total message bodies deleted 1
Archive status running
```

If errors occur during the archive operation, you see the following display.

```
Total number of errors XX [show error log]
```

[show error log] is a link that toggles with **[hide error log]**. You can click this link to show or hide the error log. The maximum number of errors displayed in the table is 1000. Each time you run an archive operation, it deletes the previous archive error log.

4.3 Default Behavior

If you use the default class (Ens.Archive.Manager), Ensemble does the following for each message to be archived:

• Copies the message header to the target namespace.

• Copies the serialized message body (not the message body object) to the target namespace.

• Deletes the message header and message body objects from the original namespace.

Note: Messages cannot be restored once archived to another namespace.

5

Managing Workflow Roles, Users, and Tasks

This chapter describes how to configure workflow users and roles. It also describes (for supervisors) how to manage workflow activity. It contains the following sections:

- Introduction to the Workflow Menu

- Managing Workflow Roles

- Managing Workflow Users

- Managing Workflow Tasks

- Viewing the Assigned Tasks

5.1 Introduction to the Workflow Menu

The Management Portal provides pages for configuring workflow users and roles and for monitoring workflow activity. To access them, click **Ensemble**, click **Manage**, and then click **Workflow**.

These pages are primarily meant for supervisors. Supervisors can assign or cancel tasks, but other actions (such as marking tasks complete) are not available here. Instead, users manage their workflow tasks within the DeepSee User Portal, which also displays Ensemble dashboards. For information, see "Using the Portal Features" in the *DeepSee End User Guide*.

5.2 Managing Workflow Roles

The **Workflow Role Profiles** page lists workflow roles currently defined in the Ensemble namespace. To display this page, click **Ensemble** > **Manage** > **Workflow** > **Workflow Roles**.

On this page, you can do the following:

- Edit the details of a role. To do so, click a role in the table. Edit the following details on the right:

    – **Name** — The role name. This is identical to the configured **Name** of the corresponding workflow operation in the production. See *Defining Workflows*.

    – **Description** — A descriptive name for the workflow role.

- **Capacity** — The maximum number of active tasks a workflow role is expected to have. This number is used in calculating performance metrics. The default is 100.

  Then click **Save**.

- Add a user to a role. To do so, click a role in the table. Then click **Add**.

  Provide the following details:

  - **Username** — Select a workflow user. The system lists all the Ensemble user IDs that have been configured as workflow users; see the next topic.

  - **Rank** — Optionally select an integer to indicate the ordinal rank of the user within this role. This value can affect task distribution. For example, you could use 1 for the more senior members of the role and 2 for the other members.

  - **Title** — Optionally specify a string that clarifies the user's job position. This value can affect task distribution. For example, a user can be designated as the "manager" of a workflow role.

  Then click **OK**.

- Remove a user from a role. To do so, click a role in the table and then click the lower **Remove** button (next to **Add**).

  Then select a user and click OK.

- See the users currently in this role. To do so, click a role in the table and then click **Users**. The system displays a table of users in a dialog box.

- See the tasks currently associated with or assigned to users in this role. To do so, click a role in the table and then click **Tasks**. The system displays a table of tasks in a dialog box.

- Remove a role. To do so, click a role in the table and then click the upper **Remove** button (next to **Save**). Click **OK** to confirm.

5.3 Managing Workflow Users

The **Workflow Users** page lists workflow users currently defined in the Ensemble namespace. To display this page, click **Ensemble** > **Manage** > **Workflow** > **Workflow Users**.

On this page, you can do the following:

- Configure an existing Ensemble user as a workflow user. To do so, click the username from the **Name** drop-down list. Optionally specify the following additional details:

  - **Description** — A descriptive name for the user.

  - **Active?** — Controls whether this user is currently active, as a workflow user.

  Then click **Save**.

- Edit the details of a username. To do so, click a username in the table. Edit the details and then click **Save**.

- See the roles to which a user belongs. To do so, click a username in the table and then click **Roles**. The system displays a table of roles in a dialog box.

- See the tasks currently associated with or assigned to this user. To do so, click a username in the table and then click **Tasks**. The system displays a table of tasks in a dialog box.

- Remove a user definition from this table. To do so, click a user and then click **Remove**.

  This does not remove the Ensemble user definition.

5.4 Managing Workflow Tasks

The **Workflow Task List** page lists all tasks that have passed through the production, since the last time messages were purged for this production. To display this page, click **Ensemble** > **Manage** > **Workflow** > **Workflow Tasks** from the Home page.

The following shows an example of this page:

The **Status** column uses the following background colors to indicate the status of the tasks:

- Yellow — Unassigned. This task is active and appears in the Worklist Inbox of each workflow user.

- Dark blue — Assigned. This task is active and appears in the Worklist Inbox of the assigned workflow user. This status does not indicate whether or not the user has accepted the task.

- Gray — Completed. This task is inactive; inactive tasks are not displayed in the Worklist Inbox of any user.

- Orange — Cancelled (a supervisor cancelled the task before it is completed). This task is inactive.

- Pink — Discarded (the request timeout period expired before the task was completed). This task is inactive.

On this page, you can do the following:

- Assign a task to a user. To do so, click the task in the table and then click **Assign Task**. Specify the following details:

  – Optionally select a different task ID from the first drop-down list.

  – Select a username from the drop-down list.

  – Optionally select a different priority from the **Priority** drop-down list.

    The **Priority** value indicates the relative priority of the task. 1 is the highest priority. A task has a default priority, but you can change this when assigning a task.

  – Optionally edit the description in the **Subject** field.

  Then click **OK**.

- Change the priority of a task without assigning it to a specific user. To do so, click the task in the table and then click **Assign Task**. Then modify the value for **Priority** and click **OK**.

- Cancel a task. To do so, click the task in the table and then click **Assign Task**. Then select **Cancel?** and click **OK**. The task is immediately canceled.

  WARNING!   You cannot undo any of the preceding operations.

- Display details for a task. To do so, click the task in the table and then click the >> symbol in that row.

- Hide the task details. To do so, click **Hide Details**.

5.4.1 Other Details

Because tasks are messages, this page lists all tasks since the last time messages were purged for this production. For details about message purging, see "Purging Management Data."

For information on how users access their Worklist Inboxes, see "Using the Portal Features" in the *DeepSee End User Guide*.

For reference, the columns in this table have the following meanings:

- **TaskId** — The **MessageId** of the task request message that the business process sends to the workflow operation.

- **RoleName** — The name of the workflow operation to which the task request was addressed.

- **Status** — Described in the preceding section.

- **Priority** — Described in the preceding section.

- **Source** — The configuration name of the business process that sent the task request to the workflow operation.

- **AssignedTo** — The workflow user to whom this task is assigned, if any.

- **Subject** — An optional text string that identifies the purpose of the task. In responses, this string is a copy of the subject value provided in the initial request for the task.

- **TimeCreated** — The date and time stamp when the Workflow Engine first received the task request and created the corresponding task response object.

- **TimeCompleted** — For inactive tasks (Completed, Discarded, or Cancelled), the date and time stamp when the workflow operation returned the completed task response object to the business process.

- **Duration** — For inactive tasks (Completed, Discarded, or Cancelled), this number is the difference, in seconds, between **TimeCreated** and **TimeCompleted**. The **Duration** value represents the amount of time the task spent inside the human workflow (that is, the amount of time during which it was visible in Workflow Inboxes).

5.5 Viewing the Assigned Tasks

The **Current Workflow Worklist** page lists all assigned tasks (that is, tasks whose status is Assigned) in the production. To display this page, click **Ensemble** > **Manage** > **Workflow** > **Workflow Worklist** from the Home page.

The following shows an example of this page:

Item ID	Task ID	User Name	Role Name	Priority	Time Created	Age	Assigned To	Subject
5||EHaskins	5	EHaskins	Demo-Development	3	2011-12-08 16:51:45.875	00w 0d 18h 11m 05s		Problem reported by ADonnelly
5||HDonovan	5	HDonovan	Demo-Development	3	2011-12-08 16:51:45.875	00w 0d 18h 11m 05s		Problem reported by ADonnelly
10||EHaskins	10	EHaskins	Demo-Development	1	2011-12-08 16:51:45.875	00w 0d 18h 11m 05s		Problem reported by Selim Jones
10||HDonovan	10	HDonovan	Demo-Development	1	2011-12-08 16:51:45.875	00w 0d 18h 11m 05s		Problem reported by Selim Jones
36||HDonovan	36	HDonovan	Demo-Testing	3	2011-12-08 15:44:11.394	00w 0d 19h 18m 39s		Test this problem from Nelson Q Jefferson
36||JDBowman	36	JDBowman	Demo-Testing	3	2011-12-08 15:44:11.394	00w 0d 19h 18m 39s		Test this problem from Nelson Q Jefferson
36||SuperUser	36	SuperUser	Demo-Testing	3	2011-12-08 15:44:11.394	00w 0d 19h 18m 39s		Test this problem from Nelson Q Jefferson
34||HDonovan	34	HDonovan	Demo-Development	3	2011-12-08 15:38:14.600	00w 0d 19h 24m 36s	HDonovan	Problem reported by Mercy H Wellbeing

The **ItemId** column is the internal identifier for the tasks. It consists of the numeric **TaskId**, the string ||, and the name of the user to which the task has been assigned.

The **Age** column indicates the elapsed time since the task response object was created. This time indicates the progress of the task towards its timeout. When the Age value exceeds the timeout for the task, the task is discarded. If there is no timeout, the task stays active until a user completes it, and this value simply increments.

The **Assigned To** column is either of the following:

- Null (if the assigned user has not yet accepted the task).

- Username to whom it was assigned (if the user has accepted it).

For information on the other columns, see the details for the **Workflow Task List** page, in the preceding section.

6

Defining Publish and Subscribe Message Routing

Ensemble supports publish and subscribe message delivery. *Publish and subscribe* refers to the technique of routing a message to one or more subscribers based on the fact that those subscribers have previously registered to be notified about messages on a specific topic. This chapter discusses the following topics:

- Publish and Subscribe Overview

- Implementing Publish and Subscribe Message Delivery

6.1 Publish and Subscribe Overview

Ensemble publish and subscribe messaging works based on the runtime interactions between:

- Messages

- Topics

- Subscribers

- Subscriptions

6.1.1 Messages

A *message* is an Ensemble message. A system that is external to Ensemble receives a request and directs it into Ensemble, which converts it to an Ensemble message and sends it to a special-purpose business operation for processing.

6.1.2 Topics

A *topic* is a string that characterizes the contents of a message. Ensemble does not define any topics; users and their applications define the meanings of topics and subtopics.

A topic string has the form `A.B.C.D`, where `A`, `B`, `C`, and `D` are *subtopic* strings delimited by the `.` (period) character. A topic can contain any number of subtopics; each of these subtopics can be up to 50 characters long. The following are all valid topic strings:

```
books
books.fiction
books.fiction.latin
```

You can specify a range of topics by using * (the asterisk) as a wildcard character. For example:

- `*` can replace any complete subtopic in the topic string (`books.*.latin` works)

- `*` does not work as a partial wildcard (`*s.fiction` does *not* work; it does not match `books.fiction`, `reviews.fiction`, or any similar string)

- A trailing `*` character matches any number of additional subtopics to the right of the last `.` (period) character in the topic string (`books.*` matches `books.fiction` and `books.fiction.latin`)

6.1.3 Subscribers

A *subscriber* is an entity (a user or an external system) that might be interested in a specific topic or set of topics. A subscriber entry specifies how that entity wishes to be contacted; that is, how Ensemble should send a message to it.

6.1.4 Subscriptions

A *subscription* associates a subscriber with a topic string.

Suppose you have three subscribers:

```
Abel
Baker
Charlie
```

And three topics with the convention that `A.B.C` represents *person.location.identifer*:

```
Doctor.ICU.88495
Patient.LAB.*
*.*.X3562564
```

In that case, you could define the following subscriptions:

Subscriber	Topic
Abel	`Doctor.ICU.88494`
Abel	`Doctor.ICU.88495`
Baker	`Doctor.ICU.88495`
Baker	`Patient.LAB.*`
Charlie	`*.*.X3562564`

This means:

- Abel is notified whenever the exact topics `Doctor.ICU.88494` or `Doctor.ICU.88495` are processed.

- Baker is notified whenever the exact topic `Doctor.ICU.88495` is processed. In addition, Baker is notified whenever any message related to patients in the lab are processed.

- Charlie is notified whenever anything related to a doctor or patient with an identifier of `X3562564` is processed.

6.2 Implementing Publish and Subscribe Message Routing

6.2.1 Creating a Publish and Subscribe Operation

To use publish and subscribe features, you must create an Ensemble production that includes an instance of the EnsLib.PubSub.PubSubOperation class.

6.2.2 Configuring Publish and Subscribe

When you configure publish and subscribe features for an Ensemble production, the basic steps are:

1. Create domains *(optional)*.

2. Create a list of subscribers.

3. Create subscriptions to associate subscribers with topics.

From the **PubSub Subscriptions** page (**Ensemble** > **Manage** > **Publish & Subscribe**), you may click **Show Domains**, **Show Subscribers**, **Show Subscriptions**, or **Create New Subscription**. The pages for domains and subscribers are similar to that for subscriptions, but each provides a different **Create** command: **Create New Subscriber** or **Create New Domain Name**.

6.3 Technical Details

Ensemble publish and subscribe messaging uses the following classes in the EnsLib.PubSub package:

Class Name	Purpose
EnsLib.PubSub.PubSubOperation	Business operation that provides publish and subscribe message routing.
EnsLib.PubSub.Request	Request class that packages requests to the PubSubOperation class. Specifies which topic and DomainName should be used to determine how the message should be routed. Optionally, the Request may also contain the message being routed, but the PubSubOperation does not need this information to return its TargetList.
EnsLib.PubSub.Response	Response class that packages responses from the PubSubOperation class. Contains a collection of Target objects called TargetList, which the calling business process consults before dispatching the message to the required destinations.
EnsLib.PubSub.Subscriber	Persistent class that represents individual subscribers. These are entities interested in being notified when certain messages arrive. The Subscriber class includes any information needed to contact the actual subscriber.
EnsLib.PubSub.Subscription	Persistent class that stores the association between a given Subscriber and a topic string.

Class Name	Purpose
EnsLib.PubSub.DomainName	Persistent class that holds the set of PubSub domain names. Domain names are optional; like namespaces, domains provide a way to keep different subscription lists separate.
EnsLib.PubSub.Utils	Utility class that provides a programmatic API for creating and deleting domains, subscribers, and subscriptions.
EnsLib.PubSub.Target	Persistent class that provides details about how to route a message to a destination outside Ensemble. The Target object has a Target property that identifies a configured business process or business operation within the current production. The Target object has an optional Address property that can specify an external address, for example an email address.

Instead of using the Management Portal, you can manipulate the objects directly using methods in the EnsLib.PubSub.Utils class.

EnsLib.PubSub.PubSubOperation does not actually send messages to subscribers; instead, it provides a mechanism to quickly find the set of interested subscribers for a given topic. It is the responsibility of a business process that calls the PubSubOperation to dispatch messages to subscribers.

At runtime, an incoming message is sent to a business process, which examines it for identifying details. Based on this analysis, the business process assigns the message a specific topic string that does not contain any wildcard characters. It then creates an EnsLib.PubSub.Request message that contains this topic string and sends it to the PubSubOperation.

The PubSubOperation uses an extremely fast search algorithm to find and return a list of all subscribers interested in this topic. The PubSubOperation returns an EnsLib.PubSub.Response message that contains a collection of EnsLib.PubSub.Target objects called TargetList. The business process iterates over this collection to dispatch the message to each EnsLib.PubSub.Target in the collection.

7

Controlling Ensemble Data Storage

This chapter describes how you can control where Ensemble stores data. Ensemble namespaces store data in Caché databases. For general information on how to control Caché database storage, see the Caché System Administration Guide. This chapter provides some supplementary information that is useful for Ensemble installations.

This chapter contains:

- Separate Databases for Routines and Globals
- Productions and Namespaces
- Where Ensemble Stores Credentials Passwords
- Where Ensemble Stores Temporary Data

For information on where Ensemble stores DICOM data, see "Configuring a DICOM Production to Control the Storage Location" in the *Ensemble DICOM Development Guide*.

7.1 Separate Databases for Routines and Globals

When you create a new namespace, you specify the databases that contain routines (the code) and the globals (the data). For new namespaces, InterSystems recommends that you specify different databases for routines and globals. Many existing namespaces use a single database to store both routines and globals. Although it is possible to split such a database into two separate ones, it is typically not worth the effort, which includes copying the routines from one database to another.

Note: Some Ensemble classes, such as Ens.Production and Ens.Rule.Rule, can be updated dynamically but are stored in the routines database. Consequently, if you are mirroring the dynamic data in an Ensemble namespace, you should include the routines database in the mirror.

You should always compile the production on the system that it is running. Although you can compile Caché code on one system and copy the database "pre-compiled" to another system, you should not attempt this with Ensemble namespaces.

7.2 Productions and Namespaces

In most cases, productions are defined and run in the same namespace, but you can use Caché package mapping to make a production class visible in a namespace other than the one it is defined in. If you use package mapping and a production

is visible in more than one namespace, you should designate only one of these namespaces to compile and run the production, You should not compile, modify, or run the production in any other namespace. If you run or modify the same production in more than one namespace it can cause failures that are hard to diagnose. Under no circumstances should you do this. If you do not use package mapping to map a database to a namespace you do not need to be concerned about this issue.

7.3 Where Ensemble Stores Credentials Passwords

When you create a namespace, Ensemble creates a separate database to store the credentials passwords for the namespace. By having the passwords in a separate database, you can encrypt this confidential account information without having to incur the overhead of encrypting all of the namespace data.

The credentials password database is created in a subdirectory of the namespace's default database for globals. The database name and subdirectory name is the name of the namespace's default database for globals with "SECONDARY" appended. For example if the default globals database is LABS then the secondary database will be called LABSSECONDARY. The credentials passwords database is protected by a resource named after the database (e.g. %DB_LABSSECONDARY) without public access. Under most conditions no user needs to have privileges to this resource. Ensemble stores the data in this database under the global ^Ens.SecondaryData.Password. See the compatibility note "New Global and Database Used to Store Credentials Passwords" in the *Ensemble Release Notes*.

Note: If you create the primary Ensemble database as a mirrored database, then the secondary database for credential passwords is automatically mirrored using the same settings as the primary database. If you add mirroring to an existing Ensemble database, then you must explicitly add mirroring to the secondary database. For information about mirroring, see the Caché High Availability Guide.

7.4 Where Ensemble Stores Temporary Data

Ensemble creates temporary data while a production is being run; this data is deleted when a production is stopped. In most cases, you do not need to be concerned with this temporary data, but in rare circumstances you may need to deal with it when recovering from error conditions. For new namespaces, Ensemble by default stores temporary data in a non-journaled database separate from CACHETEMP. The database name and subdirectory name is the name of the namespace's default database for Globals with "ENSTEMP" appended. For example if the default globals database is LABS then the temporary database will be called LABSENSTEMP. This database is be protected by the same resource as that protect the default globals database. For example, Ensemble stores the following globals in this database:

- `^CacheTemp.EnsRuntimeAppData`—contains the temporary data needed to run the production. By default, for new namespaces this global is stored in a database named by appending ENSTEMP to the name of the database used to store the namespace globals. For example, if you create a namespace TEST and specify that the globals are stored in TESTGL, this global is stored in TESTGLENSTEMP.

- `^CacheTemp.EnsJobStatus`—an entry is written when the production is started and removed when the production is stopped.

- `^CacheTemp.EnsMetrics`—contains production metrics data, such as that displayed by the production monitor.

A

Controlling Access to Management Portal Functions

This appendix describes how the Management Portal uses the predefined security roles and resources to control access to Ensemble pages and options. It contains the following sections:

- Introduction
- Predefined Ensemble Resources
- Predefined Ensemble Roles
- Default Privileges of the Predefined Ensemble Roles
- Default SQL Privileges of the Predefined Ensemble Roles
- Customizing Ensemble Security

Important: The Ensemble upgrade procedures reset these predefined resources and roles; therefore, you should customize security only by configuring resources and roles that you create yourself.

A.1 Introduction

In addition to the basic Caché roles, Ensemble contains predefined roles which you can use to control access to the functions in the Management Portal. While these built-in roles may suit most environments, you can add additional roles to customize access to pages or functions.

The following sections describe the security structure prebuilt in Ensemble. You can use this information to determine how to assign your users to roles in your environment:

For an in-depth description of Caché security, see the *Caché Security Administration Guide*, in particular, "Authorization: Controlling User Access" in the chapter "About Caché Security."

A.2 Predefined Ensemble Resources

This section describes the Ensemble predefined resources, which all begin with the `%Ens_` prefix.

- The first subsection lists resources that protect a specific activity you can perform in Ensemble.

- The second subsection lists Ensemble code and data resources.

You can view the list of predefined Ensemble resources on the **Resources** page of the Management Portal (select **System Administration** > **Security** > **Resources** from the Home page).

For an in-depth discussion of resources, see "Assets and Resources" in the *Caché Security Administration Guide*.

A.2.1 Ensemble Resources to Protect Activities

Resource	Description
`%Ens_ConfigItemRun`	Controls starting and stopping configuration items
`%Ens_DTLTest`	Grants access to the data transformation testing utility
`%Ens_Dashboard`	Grants access to the Ensemble Production Monitor
`%Ens_Deploy`	Grants access to deployment activities
`%Ens_DeploymentPkg`	Controls the creation of deployment packages
`%Ens_EventLog`	Grants access to the Event Log
`%Ens_MessageContent`	Grants access to the contents of messages
`%Ens_MessageDiscard`	Controls discarding of queued and suspended messages
`%Ens_MessageEditResend`	Grants access to edit and resend messages
`%Ens_MessageHeader`	Grants access to message header data
`%Ens_MessageResend`	Grants access to resend messages
`%Ens_MessageResubmit`	Controls resubmission of suspended messages
`%Ens_MessageSuspend`	Controls the manual suspension of messages
`%Ens_MessageTrace`	Grants access to message trace
`%Ens_MsgBank_Dashboard`	Grants access to the Enterprise Monitor Page
`%Ens_MsgBank_EventLog`	Grants access to the Message Bank Event Log
`%Ens_MsgBank_MessageContent`	Grants access to the contents of messages in the Message Bank
`%Ens_MsgBank_MessageEditResend`	Grants permission to edit and resend messages from the Message Bank
`%Ens_MsgBank_MessageHeader`	Grants access to Message Bank header data
`%Ens_MsgBank_MessageResend`	Grants permission to resend messages from the Message Bank
`%Ens_MsgBank_MessageTrace`	Grants access to the Message Bank Visual Trace
`%Ens_Portal`[*]	Grants access to the Ensemble menus in the Management Portal
`%Ens_ProductionDocumentation`	Controls the creation of production documentation
`%Ens_ProductionRun`	Controls starting and stopping productions
`%Ens_Purge`	Controls purging of Ensemble data
`%Ens_RuleLog`	Grants access to the rule log

Resource	Description
`%Ens_TestingService`	Grants access to the business host testing service
`%Ens_ViewFileSystem`	Grants access to the Finder Dialog which allows the user to view the file system

Note: * In addition to the `%Ens_Portal` resource, access to Ensemble functions in the Management Portal in any given namespace requires the Read permission on the default global database resource for that namespace. This check is enforced for access to *all* Ensemble portal pages.

A.2.2 Ensemble Resources to Protect Code and Data

Resource	Code/Data
`%Ens._AlertAdministration`	Grants access to managed alert administration.
`%Ens_Alerts`	Grants access to alert configuration and management
`%Ens_ArchiveManager`	Grants access to the Archive Manager
`%Ens_BPL`	Grants access to BPL
`%Ens_BusinessRules`	Grants access to business rules
`%Ens_Code`	Grants access to all Ensemble classes and routines
`%Ens_Credentials`	Grants access to Ensemble credentials
`%Ens_DTL`	Grants access to DTL
`%Ens_EDISchema`	Grants access to EDI schemas
`%Ens_EDISchemaAnnotation`	Grants access to the HL7 Annotation classes.
`%Ens_Jobs`	Grants access to job data
`%Ens_LookupTables`	Grants access to lookup tables
`%Ens_MsgBank`	Grants access to Message Bank status information
`%Ens_MsgBankConfig`	Grants access to Message Bank configuration
`%Ens_ProductionConfig`	Grants access to production configuration activities
`%Ens_PurgeSchedule`	Grants access to scheduling of Ensemble purge tasks
`%Ens_Queues`	Grants access to queue data
`%Ens_RecordMap`	Grants access to Ensemble record maps
`%Ens_RoutingRules`	Grants access to routing rules
`%Ens_Rules`	Grants access to all Ensemble rules
`%Ens_SequenceManager`	Grants access to the Sequence Manager
`%Ens_SystemDefaultConfig`	Grants access to system default settings
`%Ens_WorkflowConfig`	Grants access to workflow roles and users

Note: In many cases, Ensemble default behavior uses a less granular resource (like **%Ens_Code**) which protects multiple data sources including the data protected by a more specific resource (like **%Ens_BPL**). The predefined Ensemble roles and privileges use the less granular resource, but you can choose alternative roles with more selective privileges.

A.3 Predefined Ensemble Roles

Ensemble also contains a set of predefined roles, each beginning with the %EnsRole_ prefix. These are roles designed to reasonably secure your Ensemble instances in both development and production environments. The following descriptions contain an overview of the perceived job responsibilities of members of the role and how these roles relate to other Ensemble roles.

%EnsRole_Administrator

Role for a trusted and skilled Ensemble administrator. In a live or test system this is for the person able to stop, start, and configure productions; to stop and start individual configuration items; to look at all logs, messages, and queues; to purge data; to add Ensemble default system settings; and so on. This administrator has almost unlimited ability to control the Ensemble environment, but cannot change code components other than to deploy updates.

This role is intentionally distinct from Caché administrative roles and does not grant the user any Caché-level privileges.

The **%EnsRole_Administrator** role is a member of the **%EnsRole_Operator** role, and, therefore, also holds all the privileges of that role.

%EnsRole_Developer

Role for a person developing business logic, data structures, or core Ensemble code. This includes writing code in Studio, writing DTL and BPL in either Studio or using the web interface, developing routing rules, and creating custom message schemas (HL7, record mapping, XML, etc). In addition, this role allows a user to perform many administrative tasks, as the developer should have the ability to actively debug and test various options on development instances.

By default, members of the Ensemble developer role have full programming power and as such, can modify DTL, BPL, and record maps. Ensemble provides separate resources for each type of code if you want to distinguish areas of development by creating custom roles.

The **%EnsRole_Developer** role is a member of both the **%Developer** and **%EnsRole_WebDeveloper** roles. Therefore, a user assigned to this role can perform all Caché development tasks as well as the Ensemble web developer tasks.

%EnsRole_WebDeveloper

Role for a person with limited development abilities. In particular, this restricts a user to the development tasks in the Ensemble menus of the Management Portal, like BPL, DTL, defining rules, and creating record maps. The role does *not* grant access to Studio or the terminal.

This role is a member of the **%EnsRole_RulesDeveloper** and **%EnsRole_Operator** role, so that a user that is a member of this role can perform debugging tasks in the Management Portal.

%EnsRole_RulesDeveloper

> Role for a business analyst allowed to modify business rules dynamically. If you have developed a business process that requires such a function, you can allow a small number of people to modify the rules. This is not an administrative or development function.
>
> **%EnsRole_WebDeveloper** is a member of this role.

%EnsRole_Monitor

> Role for a generic user to view the Ensemble system monitor and the production monitor. Actions that would leave an audit trail if done from a user with **%EnsRole_Operator** have no effective audit trail from this generic username and therefore access needs to be restricted to a subset that does not include any risk of seeing sensitive data.

%EnsRole_Operator

> Role for operation staff managing the day-to-day status of a particular production. Users assigned to this role have the Read permission on the current configuration to determine what settings and code are in effect, but do not have permissions to modify the configuration. Operations staff may start and stop interfaces, and may start and stop the production. They do not have access to the contents of messages, but may resend messages which cause issues. Operators may view queue and job information, and may inspect the settings for purges, alerts, credentials, and lookup tables.
>
> Both **%EnsRole_Administrator** and **%EnsRole_WebDeveloper** are members of this role.

%EnsRole_AlertAdministrator

> Role for processing managed alerts assigned to any user or unassigned. For more information on processing managed alerts, see "Acting on Alerts by Viewing My Managed Alerts".

%EnsRole_AlertOperator

> Role for processing managed alerts assigned to current user or unassigned. For more information on processing managed alerts, see "Acting on Alerts by Viewing My Managed Alerts".

%EnsRole_PubSubDeveloper

> Role that allows user to control the subscription criteria used to select messages and to specify the users to receive the messages. This role provides access to the management portal page that controls Publish and Subscribe routing. For more information on Publish and Subscribe messages, see "Defining Publish and Subscribe Message Routing".

The default Ensemble security framework assigns permissions to the predefined resources, thus creating privileges for each of these roles. You can choose to assign the users of your application to these Ensemble roles or create your own roles, assigning them permissions to the Ensemble resources. If you upgrade your Ensemble instance, the procedures reset the default Ensemble roles, so you should make your configuration modifications only on user-created roles.

The next section shows the privileges assigned by default to each role.

You can view the list of predefined Ensemble roles on the **Roles** page of the Management Portal (select **System Administration** > **Security** > **Roles** from the Home page).

These roles only cover functions in the Ensemble menus of the Management Portal. Users in your environment likely require additional Caché roles. For details, see "Roles" in the *Caché Security Administration Guide*.

A.4 Default Privileges of the Predefined Ensemble Roles

This section lists the default privileges that each Ensemble role has for each Ensemble resource.

- The first subsection lists the role privileges for the activity resources.

- The second subsection lists the role privileges for the code and data resources.

See "Privileges and Permissions" in the *Caché Security Administration Guide* for an explanation of how you grant access to resources through role privileges.

A.4.1 Role Privileges for the Activity Resources

The following table lists the role privileges for the activity resources. Only the Use permission is required for access, use this permission on the underlying resource to determine access to data as well.

Resource	%EnsRole _Administrator	%EnsRole _Developer*	%EnsRole _Monitor	%EnsRole _Operator
%Ens_ConfigItemRun	Use	Use		Use
%Ens_DTLTest	Use	Use		
%Ens_Dashboard	Use	Use	Use	Use
%Ens_Deploy	Use			
%Ens_DeploymentPkg	Use	Use		
%Ens_EventLog	Use	Use		Use
%Ens_MessageContent	Use	Use		
%Ens_MessageDiscard	Use	Use		
%Ens_MessageEditResend	Use	Use		
%Ens_MessageHeader	Use	Use		Use
%Ens_MessageResend	Use	Use		Use
%Ens_MessageResubmit	Use	Use		
%Ens_MessageSuspend	Use	Use		
%Ens_MessageTrace	Use	Use		Use
%Ens_MsgBank_Dashboard	Use	Use	Use	Use
%Ens_MsgBank_EventLog	Use	Use		Use
%Ens_MsgBank_MessageContent	Use	Use		
%Ens_MsgBank_MessageEditResend	Use	Use		
%Ens_MsgBank_MessageHeader	Use	Use		Use
%Ens_MsgBank_MessageResend	Use	Use		Use
%Ens_MsgBank_MessageTrace	Use	Use		Use
%Ens_Portal*	Use	Use	Use	Use

Resource	%EnsRole _Administrator	%EnsRole _Developer*	%EnsRole _Monitor	%EnsRole _Operator
`%Ens_ProductionDocumentation`	Use	Use		
`%Ens_ProductionRun`	Use	Use		Use
`%Ens_Purge`	Use	Use		
`%Ens_RuleLog`*	Use	Use		Use
`%Ens_TestingService`	Use	Use		
`%Ens_ViewFileSystem`	Use	Use		

A.4.2 Role Privileges for the Code and Data Resources

The following table lists the role privileges for the code and data resources. Read and Write permissions are distinct for the resource; your application code should use these two permissions to determine access to the underlying data.

For reasons of space, this table does not include the information on all roles. Additional roles are described after the table.

Resource	%EnsRole _Administrator	%EnsRole _Developer	%EnsRole _Monitor	%EnsRole _Operator
`%Ens_Alerts`	Read, Write	Read, Write		Read
`%Ens_ArchiveManager`	Read, Write			
`%Ens_BPL`				
`%Ens_BusinessRules`				
`%Ens_Code`	Read	Read, Write		
`%Ens_Credentials`	Read, Write	Read		Read
`%Ens_DTL`				
`%Ens_EDISchema`	Read	Read, Write		
`%Ens_Jobs`	Read, Write	Read, Write		Read
`%Ens_LookupTables`	Read, Write	Read, Write		Read
`%Ens_MsgBank`	Read, Write	Read		Read
`%Ens_MsgBankConfig`	Read, Write	Read, Write		
`%Ens_ProductionConfig`	Read, Write	Read, Write		Read
`%Ens_PurgeSchedule`	Read, Write	Read		Read
`%Ens_Queues`	Read, Write	Read, Write		Read
`%Ens_RecordMap`				
`%Ens_RoutingRules`				
`%Ens_Rules`*		Read, Write		
`%Ens_SequenceManager`	Read, Write	Read, Write		
`%Ens_SystemDefaultConfig`	Read, Write	Read		Read
`%Ens_WorkflowConfig`	Write	Read, Write		Read

Additional roles have the following privileges

- The **%EnsRole_WebDeveloper** role has the same privileges as **%EnsRole_Developer**.

- The **%EnsRole_RulesDeveloper** role has only the following privileges:

    - **%Ens_Portal:U**

    - **%Ens_RuleLog:U**

    - **%Ens_Rules:RW**

A.4.3 Portal Page Privilege Requirements

Each Management Portal page has a default privilege requirement in the security framework shipped with Ensemble. You can view this requirement while in the columns view of the portal menu just beneath where you click **Go** to navigate to the page. You only see this information if you click next to the menu item name and not directly on the label.

For example, when you click to the right of **Production** on the Ensemble **Configure** menu of the Management Portal, you see **%Ens_ProductionConfig:READ** listed under the **System Resource(s)** label. This means you must be a member of a role that has the Read permission on the **%Ens_ProductionConfig** resource to view the **Production Configuration** page.

Notice you may also assign custom resources to a portal page. See "Using Custom Resources with the Management Portal" in the chapter "Assets and Resources" in the *Caché Security Administration Guide*.

A.5 Default SQL Privileges of the Predefined Ensemble Roles

Several Ensemble pages in the Management Portal use SQL queries to retrieve information; therefore, Ensemble users must have privileges on the appropriate tables to view this information. This section shows how Ensemble assigns SELECT privileges to its predefined roles to provide the proper security.

The **%EnsRole_Administrator**, **%EnsRole_Developer**, and **%EnsRole_WebDeveloper** roles hold the SELECT privilege on all of the following Ensemble SQL tables:

- Ens.BusinessProcess

- Ens.BusinessProcessBPL

- Ens.MessageBody

- Ens.MessageHeader

- Ens.StreamContainer

- Ens.StringContainer

- EnsLib_DICOM.Document

- EnsLib_EDI_ASTM.Document

- EnsLib_EDI_ASTM.SearchTable

- EnsLib_EDI_EDIFACT.Document

- EnsLib_EDI_EDIFACT.SearchTable

- EnsLib_EDI_X12.Document

- EnsLib_EDI_X12.SearchTable

- EnsLib_EDI_XML.Document

- EnsLib_EDI.XML.SearchTable

- EnsLib_HL7.Message

- EnsLib_HL7.SearchTable

- EnsLib_Printing.PrintJob

- EnsLib_Printing.PrintRequest

- EnsLib_SQL.Snapshot

- EnsLib_XML.SearchTable

- EnsLib_ebXML.Message

- EnsLib_ebXML.MessageTracking

- EnsLib_ebXML.MessageWithPayload

- Ens_Config.Credentials

- Ens_Enterprise_MsgBank.Log

- Ens_Enterprise_MsgBank.MessageHeader

- Ens_Enterprise_MsgBank.Node

- Ens_Rule.Log

- Ens_Rule.RuleLog

- Ens_Util.Calendar

- Ens_Util.IOLog

- Ens_Util.Log

- Ens_Util.Schedule

The remaining Ensemble roles have SELECT privileges on a subset of the Ensemble SQL tables as shown in the following table.

SQL Table Name	%EnsRole _RulesDeveloper	%EnsRole _Monitor	%EnsRole _Operator
Ens.BusinessProcess			SELECT
Ens.BusinessProcessBPL			SELECT
Ens.MessageHeader			SELECT
Ens_Config.Credentials			SELECT
Ens_Enterprise_MsgBank.Log			SELECT
Ens_Enterprise_MsgBank.MessageHeader			SELECT
Ens_Enterprise_MsgBank.Node			SELECT
Ens_Rule.Log	SELECT		SELECT
Ens_Rule.RuleLog	SELECT		SELECT
Ens_Util.Calendar			SELECT
Ens_Util.Log		SELECT	SELECT
Ens_Util.Schedule			SELECT

Ensemble also grants EXECUTE privileges on the **Ens.IsASub** stored procedure (used in certain searches of the Message Viewer) to **%EnsRole_Administrator**, **%EnsRole_Developer**, and **%EnsRole_WebDeveloper**. If you define a custom role and want a user with the role to be able to perform searches on messages, you should grant this privilege to the role or user. To see if a specific role has this privilege in an Ensemble namespace:

1. Select **System Administration**, **Security**, and **Roles**.

2. Select the role.

3. Select the **SQL Procedures** tab.

4. Select the namespace from the drop-down menu.

If the role has the Ens.IsASub privilege, Ens.IsASub is listed and marked as having **EXECUTE** privilege. If the role does not have this privilege in the namespace, you can give it this privilege by doing the following on the **SQL Procedures** tab:

1. Click the **Add Procedures ...** button.

2. Select the Ens schema from the drop-down menu.

3. Select **IsASub** from the **Available** column.

4. Click the right arrow to add **IsASub** to the **Selected** column.

5. Click **Apply** and then **Close**.

You can also give this SQL procedure privilege directly to a user.

Note: Ensemble automatically grants permissions to allow the specified roles to run SELECT statements as described in the previous tables. It grants these permissions for the tables generated for the built-in message types. If you define custom message types, you should grant the same permissions to these roles for the tables generated for these custom message types.

A.6 Customizing Ensemble Security

For information on customizing Ensemble security, see the following sections in *Caché Security Administration Guide*:

- "Using Custom Resources with the Management Portal" in the chapter "Assets and Resources"

- "Creating and Editing Users" in the chapter "Users"

- "Web Applications" in the chapter "Applications"

B

Finding Information on Ensemble Menu Items

For reference, this appendix describes where to find information for the options of the **Ensemble** menu of the Management Portal. It includes the following topics:

- Configure Menu
- Build Menu
- View Menu
- List Menu
- Monitor Menu
- Manage Menu
- Interoperate Menu
- Test Menu

Also see the appendix "Controlling Access to Management Portal Functions."

For information on other menus of the Management Portal, see the *Caché System Administration Guide*.

B.1 Configure Menu

Every item on this menu is described in *Configuring Ensemble*, in the following sections:

Option	Purpose	See
Production	View, edit, start, or stop a production	"Creating and Configuring a Production"
Business Partners	Create, view, or edit business partners	"Defining Business Partners"
Credentials	Create, view, or edit credentials	"Defining Credentials"
Schedule Specs	Create, view, or edit schedule specifications	"Defining Schedule Specifications"

Option	Purpose	See
Data Lookup Tables	Create, view, or edit lookup table settings	"Defining Data Lookup Tables"
System Default Settings	Create, view, or edit system-side configuration default values	"Defining Production Defaults"
Message Bank Link	Configure the link to the Enterprise Message Bank	"Configuring the Enterprise Message Bank"

B.2 Build Menu

The following table briefly describes the options on the **Build** menu and indicates where to find information.

Option	Purpose	See
Business Processes	Create, view, or edit business processes	*Developing BPL Processes*
Data Transformations	Create, view, or edit data transformations	*Developing DTL Transformations*
Business Rules	Create, view, or edit business rules	"Creating and Editing Rule Sets" in *Developing Business Rules*
Record Maps	View or edit file format record maps	"Creating a Record Map" in *Developing Ensemble Productions*
CSV Record Wizard	Create record maps from delimited files	"Using the CSV Record Wizard" in *Developing Ensemble Productions*

B.3 View Menu

Every item on this menu is described in *Monitoring Ensemble*, in the following sections:

Option	Purpose	See
Messages	View or search messages	"Viewing, Searching, and Managing Messages"
Suspended Messages	Manage suspended messages	"Managing Suspended Messages"
Event Log	View or search the Event Log	"Viewing the Event Log"
Business Rule Log	View or search the rule log	"Viewing the Business Rule Log"
Business Process Log	View or search business process instances	"Viewing Business Process Instances"
Enterprise Message Bank	Go to the Enterprise Message Bank / Monitor portal	"Using the Enterprise Message Bank"

B.4 List Menu

The following table briefly describes the options on the **List** menu and indicates where to find information.

Option	Purpose	See
Business Processes	View a list of business processes	"Business Process List" in *Developing BPL Processes*
Data Transformations	View a list of data transformations	*Developing DTL Transformations*
Business Rules	View a list of business and routing rules	"Business Rule List" in *Developing Business Rules*
Record Maps	View a list of record maps	"Creating a Record Map" in *Developing Ensemble Productions*
Productions	Manage other productions	"Viewing the Production List" in *Configuring Ensemble Productions*

B.5 Monitor Menu

Every item on this menu is described in *Monitoring Ensemble*, in the following sections:

Option	Purpose	See
System Monitor	View the Ensemble system monitor to monitor productions in all namespaces	"Using the Ensemble System Monitor"
Production Monitor	Monitor a single production more closely	"Monitoring a Production"
Queues	View Ensemble queues	"Monitoring Production Queues"
Jobs	View Ensemble jobs	"Monitoring Active Jobs"

B.6 Manage Menu

Every item on this menu is described in *this* book, in the following sections:

Option	Purpose	See
Purge Management Data	Purge messages, logs, and monitor records	"Purging Management Data"
Auto-start Production	Select a production to start and stop automatically	"Managing Production Auto-start"
Local Archive Manager	Define or run local archives	"Using the Local Archive Manager"

Option	Purpose	See
Workflow	Create, view, or edit workflow roles, users, tasks, and worklists	"Managing Workflow Roles, Users, and Tasks"
Publish & Subscribe	Manage publish and subscribe message delivery	"Defining Publish and Subscribe Message Routing"

B.7 Interoperate Menu

The Ensemble **Interoperate** menu options provide access to tasks you perform to view and transform Electronic Data Interchange (EDI) messages to determine how you want your Ensemble production to handle them. To display the page, click **Interoperate** in the Ensemble menu.

The following table briefly describes the options on the **Interoperate** menu and indicates where to find information.

Option	Purpose	See
ASC X12	View, import, and remove X12 schemas; view and transform X12 documents	"Available Tools" in the *Ensemble X12 Development Guide*
ASTM	View, import, and remove ASTM schemas; view and transform ASTM documents	"Available Tools" in the *Ensemble ASTM Development Guide*
DICOM	Create, view or edit DICOM associations; view DICOM abstract syntax; view DICOM dictionary	*Ensemble DICOM Development Guide*
HL7 v2.x	View, import, and remove HL7 v2.x schemas; view and transform HL7 v2.x documents	"Available Tools" in the *Ensemble HL7 Version 2 Development Guide*
UN/EDIFACT	View, import, and remove EDIFACT schemas; view and transform EDIFACT documents	"Available Tools" in the *Ensemble EDIFACT Development Guide*
XML	View, import, export, and remove XML schemas; view and transform XML documents	"Available Tools" in the *Ensemble XML Virtual Document Development Guide*

B.8 Test Menu

The following table briefly describes the options on the **Test** menu and indicates where to find information.

Option	Purpose	See
Business Hosts	Run the testing service for business processes or business operations	"Using the Testing Service" in *Developing Ensemble Productions*
Data Transformations	View the results of data transformations on sample messages	*Developing DTL Transformations*