Managing Productions

Version 2023.1
2024-07-11

InterSystems Corporation 1 Memorial Drive Cambridge MA 02142 www.intersystems.com

Managing Productions
InterSystems IRIS Data Platform Version 2023.1 2024-07-11

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

1

Introduction to Managing Productions

This topic introduces the Management Portal and the tasks involved in managing productions in InterSystems IRIS® data platform.

1.1 Background for System Administrators

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

A *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.

InterSystems IRIS 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.

1.2 Introduction to Managing Productions

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

- Purging old data that is no longer needed.

  InterSystems IRIS stores messages, Event Log entries, business rule log entries, and other historical data. It is generally necessary to purge old data periodically. For guidance, see Purging Data.

- Enabling or disabling the auto-start option for productions. This option causes a production to start automatically when InterSystems IRIS starts, and to shut down when InterSystems IRIS is stopped.
- Creating and maintaining workflow roles and users, if any productions use the InterSystems IRIS 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.
- 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 Productions.

1.3 Introduction to the Management Portal

In the Management Portal, the **Interoperability** menu provides options that apply specifically to productions.

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.

1.4 Getting Started with the Management Portal

1.4.1 Accessing the Management Portal

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

- Select the InterSystems launcher  in the Windows system tray and click **Management Portal**.
- Use a previously saved bookmark to the Management Portal page.

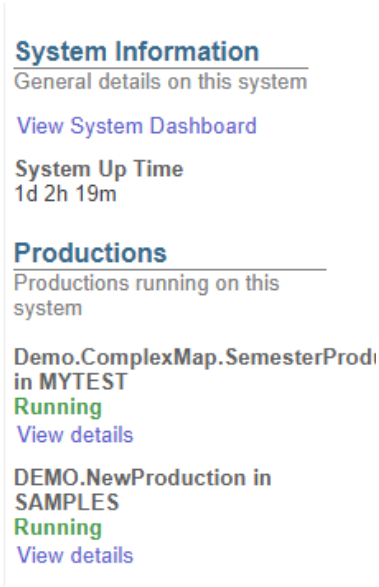
1.4.2 Choosing a Namespace

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

When you choose the **Interoperability** menu, if you have not selected an interoperability-enabled namespace, you are prompted to switch to a different namespace. Select an interoperability-enabled namespace. For information about namespaces in InterSystems IRIS, see Environmental Considerations.

1.5 Viewing Summaries for Active Productions

When you click any option in the **Interoperability** 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.

1.6 See Also

- Finding Information on Menu Items
- Management Portal Overview
- Management Portal Page Reference

2

Starting and Stopping Productions

By default, InterSystems IRIS® does not automatically start productions. This topic describes how to start and stop productions.

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.

Task	Privilege Needed
Access Interoperability menus in Management Portal	• **%Ens_Portal:USE** • READ permission on the default global database resource for the namespace
Start or Stop a Production	• **%Ens_ProductionRun:USE**
Manage Relative Startup Priority	• READ permission on the default global database resource for *all* namespaces
Manage Deployment and Create Deployment Packages	• **%Ens_Deploy:USE** • **%Ens_DeploymentPkg:USE** • **%Ens_DeploymentPkgClient:USE**

2.1 Starting a Production

To start a production from the Management Portal:

1. Select **Interoperability** > **List** > **Productions**.

   InterSystems IRIS then displays the Production List page.

2. Select the production that you want to start.
3. Select Open.

   InterSystems IRIS displays the production.

4. Select **Start**.
5. Select **OK**. InterSystems IRIS 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, select **OK**.

2.2 Stopping a Production

To stop a production from the Management Portal:

1. Select **Interoperability** > **List** > **Productions**.

   InterSystems IRIS then displays the Production List page.
2. Select the production that you want to stop. This must be a production that is running.
3. Select Open.

   InterSystems IRIS displays the production.
4. Select **Stop**.
5. Select **OK**. InterSystems IRIS 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, select **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.

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.

If you have access to all namespaces, you can assign a Relative Startup Priority to an auto-start production. When the system starts up, the production with the highest priority number starts first, regardless of its namespace. If two productions share a priority number, the alphabetical order of the productions' namespaces determines which production starts first. You cannot set a Relative Startup Priority if you do not have access to all namespaces.

To access this page in the Management Portal, select **Interoperability** > **Manage** > **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. If you have access to all namespaces, set the **Relative Startup Priority**. Productions with the highest number start first.
3. Click **Apply**.

InterSystems IRIS 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**.

InterSystems IRIS 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.

2.4 Overriding the Production Auto-Start Options in All Namespaces

For debugging purposes or during disaster recovery, you can override the auto-start options for all productions. To do so:

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

   The **Startup Settings** page appears.
2. Select **Edit** next to the **EnsembleAutoStart** setting.
3. Clear the check box.
4. Click **Save**.

InterSystems IRIS subsequently disregards the namespace-specific settings in the **Auto-Start Production** page, which is described in the previous section. In other words, even if a production appears in the **Startup sequence priorities for productions set to Auto-Start** list for a given namespace, the system does not automatically restart the production.

2.5 Improving Restarts for Productions with Large Queues

By default, when a production is stopped, any asynchronous messages on the ^*Ens.Queue* global queue are moved to the ^*Ens.Suspended* queue. When the production is restarted, they are moved back. For productions with many messages in the queue, this can slow down the process of stopping and restarting a production. To avoid moving the messages, you can set the following ^*Ens.Configuration* global node:

```
set ^Ens.Configuration("Queues","KeepInQueues")=1
```

By default, the node is set to 0; you will have to change this for each namespace. This will prevent the messages from being moved out of the *Ens.Queue* global, which can improve restart speeds on productions which consistently have large queues.

2.6 Using Production Shutdown Groups

Production shutdown groups allow you to control the order in which productions are stopped when you shut down an instance. By default, all productions are shut down in parallel when you stop an instance. When productions are organized into production shutdown groups, InterSystems shuts down the first group of productions before starting to shut down productions in the next group. Each group name must be an integer, and InterSystems starts shutting down the lowest group number first. By default, all productions belong to group 2.

To add a production to a production shutdown group:

1. Select the production's namespace and then go to **Interoperability** > **Manage** > **Configuration** > **Production Shutdown Groups**.
2. Select the production from the drop-down list.
3. Enter the group's number in **Relative Shutdown Group**.
4. Select **Apply**.

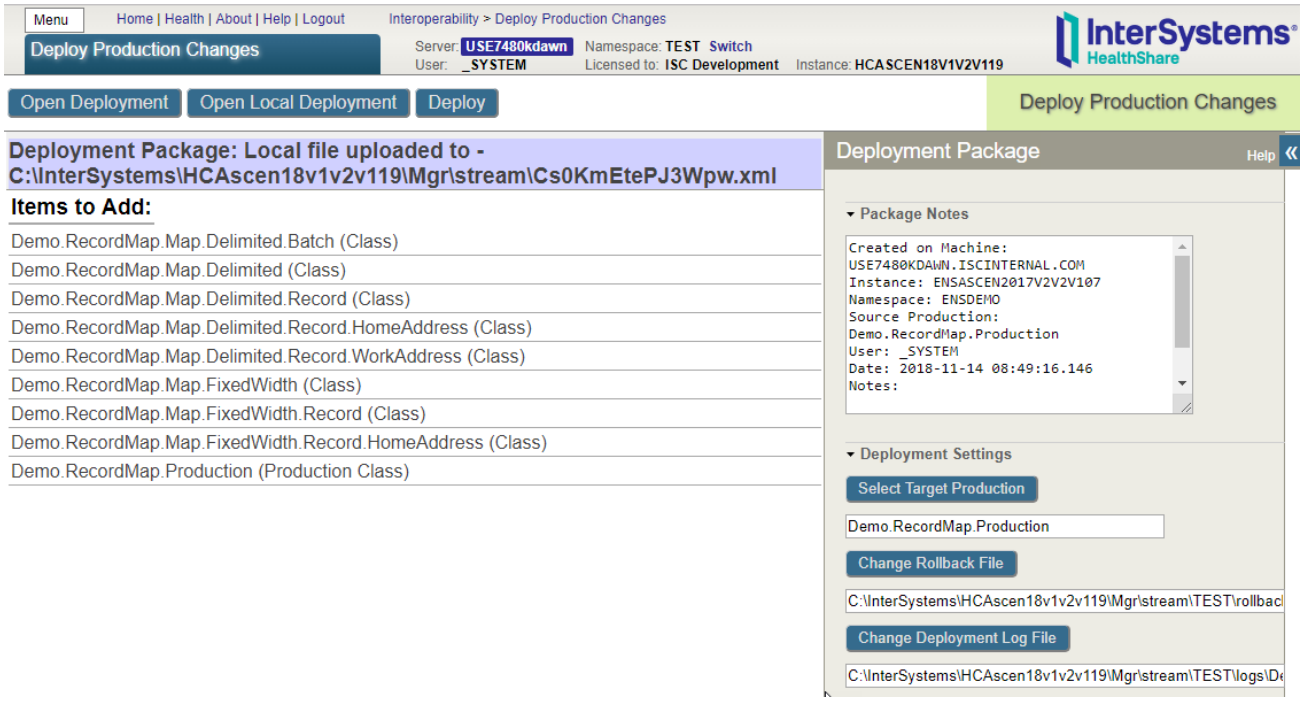
The table at the bottom of the page displays the productions that are currently active in each namespace, along with their group number. You can assign a production to a group at any time, but it does not appear in the bottom table unless it is the active production, which means it is running or is the most recently stopped production in the namespace.

Note: Using production shutdown groups can result in it taking longer for the instance to stop, which has implications for time-sensitive failovers.

2.7 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 InterSystems IRIS 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 **Interoperability**, **Manage**, **Deployment Changes**, **Deploy**, and then click the **Open Deployment** or **Open Local Deployment** button, depending on whether the XML deployment package is located on the server or on the local machine. The **Open Local Deployment** button is not active if you are on the server machine. After you select the XML deployment package file, the form lists the new and changed items in the deployment package and 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, InterSystems IRIS 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. InterSystems IRIS does the following to stop the production, load the new code, and then restart the production.

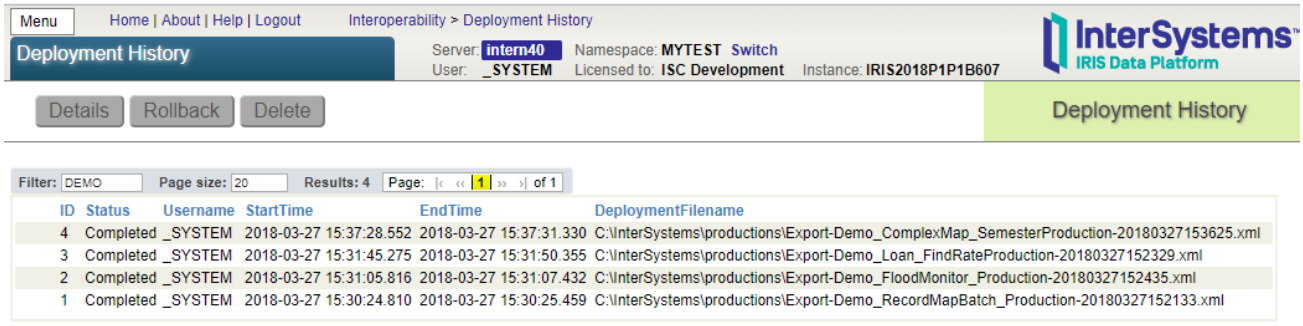
1. Create and save the rollback package.
2. Disable the components in the production that have a production settings (ptd) file in the deployment package.
3. Import the XML file and compile the code. If there is an error compiling any component, the entire deployment is rolled back.
4. Update the production 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, use the **Open Deployment** button to select the rollback file, 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.

2.8 Displaying the Deployment History

You can view the deployment history of the productions in a namespace. To view the deployment history, select **Interoperability**, **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

This page describes how and why to purge production data.

3.1 Introduction

For each production running in a given namespace, InterSystems IRIS may write entries to the event log, message warehouse, business process log, business rule log, and I/O archive log for the namespace. Since the entries can accumulate over time and consume large amounts of disk space, InterSystems IRIS enables you to purge outdated entries if you have appropriate permissions (see Controlling Access to Management Portal Functions).

You can do so manually; that is, you can purge production data on an ad hoc basis. You can also schedule regular purges. Typically, you perform manual purges on systems that you are using for development and testing, and you set up scheduled purges for live systems.

Purging generates journaling. If you purge a large volume of data, the resultant journaling can consume a large amount of disk space. To conserve disk space, you can purge smaller amounts of data and review the storage impacts before purging additional data.

3.2 First-time Purges

Purging generates journaling. If you purge a large volume of data, the resultant journaling can consume a large amount of disk space. To conserve disk space, you can adopt the following approach the first time you purge management data:

1. Switch to the namespace where you want to purge data.
2. Navigate to the **Interoperability** > **Manage** > **Purge Management Data** page.
3. Set the purge parameters so that a relatively small amount of data is purged.

   For example, you can set **Do not purge most recent** to a relatively large number. For more information, see Settings for Purging Data.

   CAUTION: Purges are irreversible and can lead to unintentionally orphaned data or the loss of unresolved requests. Consequently, InterSystems recommends that you carefully review the description of each setting before proceeding.

4. Click **Start Purge**.

5. Gradually decrease the **Do not purge most recent** value and purge additional data until you have purged a sufficient amount of data.

3.3 Purging Data Manually

The **Purge Management Data** page enables you to purge entries in the event log, message warehouse, business process log, business rule log, and I/O archive log all at one time for a given namespace. The page displays information about the entries in a table with the following columns:

- **Record Type** — Indicates the type of production data associated with the row. 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** — Shows the total number of entries of a given **Record Type** stored for the production. You can use the **Count** value to decide whether it is worthwhile to purge the entries and if so, how many days' worth of records to keep.
- **Deleted** — After you click **Start Purge** and the system completes the purge process, shows the total number of entries of a given **Record Type** that were purged.

Additionally, the **Purge Criteria** area displays the default settings that your system administrator configured for manual purges.

To purge production data manually, do the following:

1. Switch to the namespace where you want to purge data.
2. Navigate to the **Interoperability** > **Manage** > **Purge Management Data** page.
3. If you have appropriate permissions, modify the settings in the **Purge Criteria** area as needed.

   For more information, see Settings for Purging Data.

   CAUTION: Purges are irreversible and can lead to unintentionally orphaned data or the loss of unresolved requests. Consequently, InterSystems recommends that you carefully review the description of each setting before proceeding.

4. Click **Start Purge**.

   The system immediately purges the persistent store using the settings in the **Purge Criteria** area. The page uses a background job to perform 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. After the purge, the **Deleted** column displays the number of records that were purged.

   The **Start Purge** button is disabled while a purge is being executed in a given namespace.

5. Purging using the Management Portal only purges 500 chunks of bitmaps at a time. If you are purging a large number of messages this will leave unpurged bitmaps which can take up space. To remove these you can run another purge via the Terminal. Run the purge using the following settings:

```
set pDaysToKeep=7
set pKeepIntegrity=0
set pBodiesToo=1
set pBitmapChunkLimit=10000000000

write ##class(Ens.Util.MessagePurge).Purge(pDeletedCount, pDaysToKeep, pKeepIntegrity, pBodiesToo,
 pBitmapChunkLimit, pExtendedOptions)

zwrite pDeletedCount
```

3.4 Purging Data on a Schedule

The **Task Scheduler Wizard** enables you to schedule purges for the following types of production data separately or all at one time for a given namespace:

- Events
- Messages
- Business processes
- Rule logs
- I/O logs
- Host monitor data
- Managed alerts

To purge data automatically at regular intervals, do the following:

1. Navigate to **System Operation** > **Task Manager**, and then select **New Task**.
2. Fill in the following fields:
   - **Task name** — Specify a name for the purge task.
   - **Namespace to run task in** — Select the namespace where you want to purge data.
   - **Task type** — Select Ens.Util.Tasks.Purge.

   Various settings for purging data appear.
3. Modify the setting for purging data as needed.

   For more information, see Settings for Purging Data.

   CAUTION: Purges are irreversible and can lead to unintentionally orphaned data or the loss of unresolved requests. Consequently, InterSystems recommends that you carefully review the description of each setting before proceeding.
4. Specify other options as needed.

   For more information, see Using the Task Manager.
5. Click **Finish**.

3.5 Settings Applicable to Data Purges

This section describes the settings that affect data purges. These settings are all specific to the namespace in which they are set. Users with appropriate permissions can modify these.

Include message bodies

Alternative name: **BodiesToo**

Where to modify this setting:

- **Purge Management Data** page
- **Purge Data Settings** page
- **Task Scheduler Wizard** page (as **BodiesToo**)

Default: disabled

Specifies whether to purge message bodies in addition to message headers (which are always purged) during a purge operation.

If this setting is enabled, InterSystems IRIS purges message headers and their corresponding message bodies. If this setting is disabled, InterSystems IRIS purges only message headers and retains any corresponding message bodies.

The system verifies that body classes exist and are persistent before purging them.

Important: If InterSystems IRIS purges only message headers, the system may accumulate large quantities of message bodies. You cannot delete the retained message bodies from the Management Portal. You can delete them only programmatically. Consequently, InterSystems recommends that you consider your disk space and workflow when you configure the **Include message bodies** or **BodiesToo** setting.

Additionally, when InterSystems IRIS purges a message body, it does not necessarily delete all the object-valued properties of the message body. The system deletes only objects that have a serial or child relationship to the message body. You must delete other objects manually by defining a delete trigger or implementing the **%OnDelete()** method in the message body class, as appropriate. For more information about object-valued properties, see Defining and Using Object-Valued Properties.

Purge only completed sessions

Alternative name: **KeepIntegrity**

Where to modify this setting:

- **Purge Management Data** page
- **Purge Data Settings** page
- **Task Scheduler Wizard** page (as **KeepIntegrity**)

Default: enabled

Specifies whether to skip messages that are part of incomplete sessions during the purge process.

If this setting is enabled, when InterSystems IRIS encounters a message that meets the age criterion for purging, but is in an *incomplete* session, the system does not purge the message header or body. An *incomplete* session corresponds to any session that includes a message with a status other than **Complete**, **Error**, **Aborted**, or **Discarded**.

If you enable the **Purge only completed sessions** or **KeepIntegrity** setting, InterSystems IRIS executes a query that reviews all the messages (including business process instances) in each relevant session to identify any incomplete sessions. Consequently, enabling this setting can increase the amount of time required to complete a purge operation.

Preserving session-level integrity supports long-running business processes. InterSystems recommends that you consider whether you need to support long-running business processes and whether your system contains insignificant old messages in incomplete sessions when you configure the **Purge only completed sessions** or **KeepIntegrity** setting.

CAUTION: Purge operations can include messages associated with long-running system processes, such as workflow processes. If you disable this setting, carefully review the **Do not purge most recent** value to ensure that you do not purge critical system data.

Description

Where to modify this setting:

- **Purge Management Data** page
- **Purge Data Settings** page

This setting is shown in the same places as **Include message bodies** and **Purge only completed sessions** and is intended to explain those settings.

Default: "Include message bodies" is OFF because some Productions may use message objects that are part of a larger environment and not transitory. "Purge only completed sessions" is ON to preserve messages not yet completely processed.

Edit **Description** as needed, if you modify the two settings it describes.

Do not purge most recent

Alternative name: **NumberOfDaysToKeep**

Where to modify this setting:

- **Purge Management Data** page
- **Purge Data Settings** page
- **Task Scheduler Wizard** page (as **NumberOfDaysToKeep**)

Default: 7

Specifies how many days' worth of records to keep. The count of days includes *today*.

If you set the value to 0 (zero), InterSystems IRIS does not keep any records and purges all the entries that exist at the time of the purge operation. If you set the value to **1** , InterSystems IRIS retains only the messages generated on the current day, according to local server time.

TypesToPurge

Where to modify this setting:

- **Task Scheduler Wizard** page

Default: `Events`

Specifies the types of records to purge.

4

Using the Archive Manager

The **Interoperability** > **Manage** > **Local 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. Also see Using the Enterprise Message Bank

To navigate to the Archive Manager page, click **Interoperability**, click **Manage**, and then click **Local Archive Manager**. This topic describes how to use this page.

4.1 Archive Basics

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

- **Archive to namespace** — The namespace to which InterSystems IRIS® 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.
- **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 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 interoperability-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 interoperability-enabled namespaces, see Environmental Considerations.

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, the system 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), InterSystems IRIS 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 topic describes how to configure workflow users and roles. It also describes (for supervisors) how to manage workflow activity.

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, select **Interoperability** > **Manage** > **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 InterSystems User Portal, which also displays production-related dashboards. For information, see Using the Portal Features.

5.2 Managing Workflow Roles

The **Workflow Role Profiles** page lists workflow roles currently defined in the namespace. To display this page:

Select **Interoperability** > **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 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 **Interoperability** > **Manage** > **Workflow** > **Workflow Users** page lists workflow users currently defined in the namespace.

On this page, you can do the following:

- Configure an existing 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**.

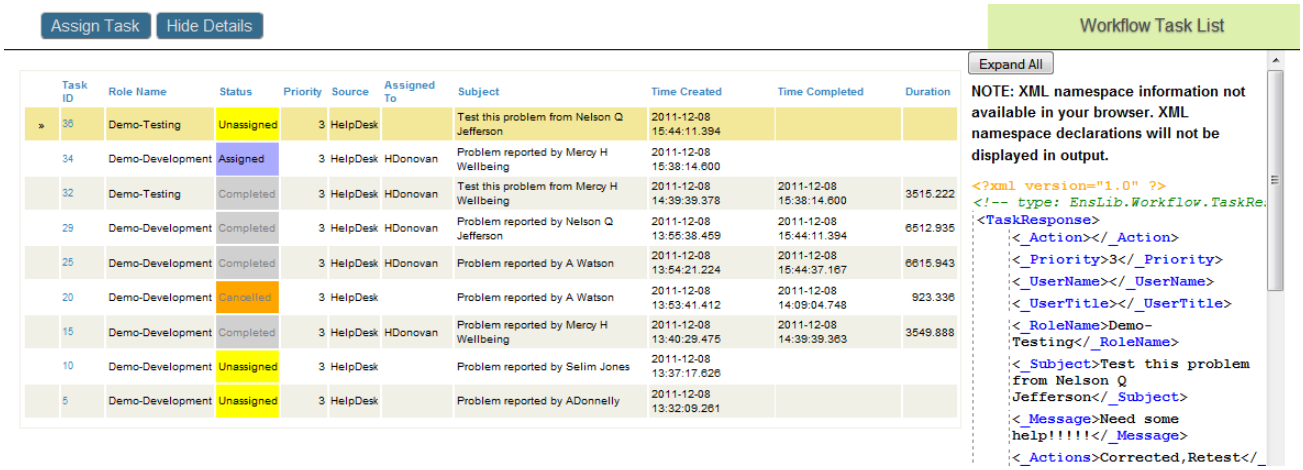
- Edit the details of a username. To do so, click a username in the table. Edit the details and then click **Save**.
- See the roles to which a user belongs. To do so, click a username in the table and then click **Roles**. The system displays a table of roles in a dialog box.
- See the tasks currently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

5.4 Managing Workflow Tasks

The **Interoperability** > **Manage** > **Workflow** > **Workflow Tasks** page lists all tasks that have passed through the production, since the last time messages were purged for this production.

The following shows an example of this page:



	Task ID	Role Name	Status	Priority	Source	Assigned To	Subject	Time Created	Time Completed	Duration
»	38	Demo-Testing	Unassigned	3	HelpDesk		Test this problem from Nelson Q Jefferson	2011-12-08 15:44:11.394		
	34	Demo-Development	Assigned	3	HelpDesk	HDonovan	Problem reported by Mercy H Wellbeing	2011-12-08 15:38:14.600		
	32	Demo-Testing	Completed	3	HelpDesk	HDonovan	Test this problem from Mercy H Wellbeing	2011-12-08 14:39:39.378	2011-12-08 15:38:14.600	3515.222
	29	Demo-Development	Completed	3	HelpDesk	HDonovan	Problem reported by Nelson Q Jefferson	2011-12-08 13:55:38.459	2011-12-08 15:44:11.394	6512.935
	25	Demo-Development	Completed	3	HelpDesk	HDonovan	Problem reported by A Watson	2011-12-08 13:54:21.224	2011-12-08 15:44:37.167	6615.943
	20	Demo-Development	Cancelled	3	HelpDesk		Problem reported by A Watson	2011-12-08 13:53:41.412	2011-12-08 14:09:04.748	923.336
	15	Demo-Development	Completed	3	HelpDesk	HDonovan	Problem reported by Mercy H Wellbeing	2011-12-08 13:40:29.475	2011-12-08 14:39:39.363	3549.888
	10	Demo-Development	Unassigned	3	HelpDesk		Problem reported by Selim Jones	2011-12-08 13:37:17.626		
	5	Demo-Development	Unassigned	3	HelpDesk		Problem reported by ADonnelly	2011-12-08 13:32:09.261		

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.

  CAUTION: 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.

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 **Interoperability** > **Manage** > **Workflow** > **Workflow Worklist** page lists all assigned tasks (that is, tasks whose status is Assigned) in the production.

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 **Interoperability** > **Manage** > **Workflow** > **Workflow Tasks** page, in the preceding section.

6

Defining Publish and Subscribe Message Routing

InterSystems IRIS® 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.

6.1 Publish and Subscribe Overview

Publish and subscribe messaging works based on the runtime interactions between:

- Messages
- Topics
- Subscribers
- Subscriptions

6.1.1 Messages

A *message* is a production message. An external system receives a request and directs it into InterSystems IRIS, which converts it to a production 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. InterSystems IRIS 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 InterSystems IRIS 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 a 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 a 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 **Interoperability** > **Manage** > **Publish & Subscribe** page, you may select **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

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 a production. 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 Data Storage for Productions

This topic describes how you can control where InterSystems IRIS® stores data. Interoperability-enabled namespaces store data in InterSystems IRIS databases. For general information on how to control InterSystems IRIS database storage, see System Administration Guide. This topic provides some supplementary information that is useful for InterSystems IRIS installations.

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 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 interoperability-enabled 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 InterSystems IRIS code on one system and copy the database “pre-compiled” to another system, you should not attempt this with interoperability-enabled namespaces.

7.2 Productions and Namespaces

In most cases, productions are defined and run in the same namespace, but you can use InterSystems IRIS 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 InterSystems IRIS Stores Password Credentials

InterSystems IRIS creates a dedicated database for password credentials when you create a new namespace with the following options enabled:

- **The default database for Globals in this namespace is a Local Database**
- **Enable namespace for interoperability productions**

Note: InterSystems IRIS never creates a password database for the USER namespace.

Additionally, InterSystems IRIS for Health and HealthShare do not create password databases by default. You can call the **CreateNewDBForSecondary()** method of the %Library.EnsembleMgr class to create them as needed.

The password database appears in a subdirectory of the directory that contains the default database for globals. Both the password database and corresponding subdirectory are named by appending SECONDARY to the name of the default database for globals. For example, if the default database for globals is named LABS, then the password database and corresponding subdirectory are named LABSSECONDARY.

InterSystems IRIS protects the database with a resource named %DB_*database*, where *database* is the name of the password database. For example, the LABSSECONDARY database is protected by the %DB_LABSSECONDARY resource. Typically, users do not require privileges to the resource protecting a password database.

The data in the password database is stored in the ^Ens.SecondaryData.Password global.

By storing passwords in a separate database, InterSystems IRIS enables you to encrypt confidential account information without having to incur the overhead of encrypting all of the namespace data.

Note: If you create the primary InterSystems IRIS database as a mirrored database, then any password database is automatically mirrored using the same settings as the primary database. If you add mirroring to an existing InterSystems IRIS database, then you must explicitly add mirroring to the password database. For information about mirroring, see High Availability Guide.

7.4 Where InterSystems IRIS Stores Temporary Production Data

While a production is running, InterSystems IRIS creates temporary data. This data is deleted when the production is stopped. While you can typically ignore temporary data, you may find it useful for recovering from an error condition.

When you create a new namespace with the following options enabled, InterSystems IRIS creates an additional, non-journaled database for temporary data:

- **The default database for Globals in this namespace is a Local Database**
- **Enable namespace for interoperability productions**

Note: InterSystems IRIS never creates databases for temporary data for the USER namespace.

InterSystems IRIS for Health and HealthShare do not create databases for temporary data by default. You can call the **createNewDBForEnsTemp()** method of the %Library.EnsembleMgr class to create them as needed.

The database for namespace-level temporary data is separate from the IRISTEMP database and contains the following globals:

- ^IRIS.Temp.EnsRuntimeAppData—Includes the temporary data required to run the production.
- ^IRIS.Temp.EnsJobStatus—Includes an entry each time a production is started, which is removed when a production is stopped.
- ^IRIS.Temp.EnsMetrics—Includes production metrics similar to the metrics displayed by the production monitor.

The database for temporary data appears in a subdirectory of the directory that contains the default database for globals. Both the temporary data database and corresponding subdirectory are named by appending ENSTEMP to the name of the default database for globals. For example, if the default database for globals is named LABS, then the temporary database and corresponding subdirectory are named LABSENSTEMP.

InterSystems IRIS protects the database with the same resource that protect the default database for globals.

A

Controlling Access to Management Portal Functions

This page describes how the Management Portal uses the predefined security roles and resources to control access to pages and options related to production management.

Note: InterSystems recommends that you do not modify predefined roles. Rather, create a new role based on the predefined role and modify the role that you have created.

A.1 Introduction

InterSystems IRIS® data platform 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 InterSystems IRIS. You can use this information to determine how to assign your users to roles in your environment.

For an overview of InterSystems security, see About InterSystems Security, and in particular, Authorization: Controlling User Access

A.2 Predefined Resources

This section describes the predefined resources related to productions. The names of these resources all begin with the `%Ens_` prefix.

- The first subsection lists resources that protect a specific activity you can perform in InterSystems IRIS.
- The second subsection lists code and data resources.

You can view the list of predefined resources on the **System Administration** > **Security** > **Resources** page of the Management Portal.

For an in-depth discussion of resources, see Assets and Resources.

A.2.1 Resources to Protect Activities Related to Productions

%Ens_AlertAdministration

Controls access to managed alert administration.

%Ens_ConfigItemRun

Controls starting and stopping configuration items.

%Ens_DTLTest

Controls access to the data transformation testing utility.

%Ens_Dashboard

Controls access to the Production Monitor.

%Ens_Deploy

Controls access to deployment activities.

%Ens_DeploymentPkg

Controls the creation of deployment packages using the server.

%Ens_DeploymentPkgClient

Controls the creation and import of local deployment packages using the browser.

%Ens_EventLog

Controls access to the Event Log.

%Ens_MessageContent

Controls access to the contents of messages.

%Ens_MessageDiscard

Controls discarding of queued and suspended messages.

%Ens_MessageEditResend

Controls access to edit and resend messages.

%Ens_MessageExport

Controls access to export messages.

%Ens_MessageHeader

Controls access to message header data.

%Ens_MessageResend

Controls access to resend messages.

%Ens_MessageSuspend

Controls the manual suspension of messages.

%Ens_MessageTrace

Controls access to message trace.

%Ens_MsgBank_Dashboard

Controls access to the Enterprise Monitor.

%Ens_MsgBank_EventLog

Controls access to the Message Bank Event Log.

%Ens_MsgBank_MessageContent

Controls 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

Controls access to Message Bank header data.

%Ens_MsgBank_MessageResend

Grants permission to resend messages from the Message Bank.

%Ens_MsgBank_MessageTrace

Controls access to the Message Bank Visual Trace.

%Ens_Portal

Controls access to the Interoperability menus in the Management Portal.

Note: To access *any* of the Interoperability pages and functions in the Management Portal for a given namespace, a user must also have Read permission on the default global database resource for the namespace.

%Ens_ProductionDocumentation

Controls the creation of production documentation.

%Ens_ProductionRun

Controls starting and stopping productions.

%Ens_Purge

Controls purging of production-related data.

%Ens_RuleLog

Controls access to the Rule Log.

%Ens_TestingService

Controls access to the business host testing service.

%Ens_ViewFileSystem

Controls access to the Finder Dialog, which enables users to browse the file system.

A.2.2 Resources to Protect Code and Data Related to Productions

%Ens_Agents

Controls access to the Agent Management page, which is applicable only to HealthShare.

%Ens_Alerts

Controls access to alert configuration and management.

%Ens_ArchiveManager

Controls access to the Archive Manager.

%Ens_BPL

Controls access to the Business Process Language (BPL).

%Ens_BusinessRules

Controls access to business rules.

%Ens_Code

Controls access to all Interoperability classes and routines.

%Ens_Credentials

Controls access to production credentials.

%Ens_DTL

Controls access to the Data Transformation Language (DTL).

%Ens_EDISchema

Controls access to EDI schemas.

%Ens_EDISchemaAnnotation

Controls access to the HL7 Annotation classes.

%Ens_ITK

Controls access to the Interoperability Toolkit, which is applicable only to HealthShare.

%Ens_JBH

Controls access to Java Business Hosts.

%Ens_Jobs

Controls access to job data.

%Ens_LookupTables

Controls access to lookup tables.

%Ens_MsgBank

Controls access to Message Bank status information

%Ens_MsgBankConfig

Controls access to Message Bank configuration.

%Ens_PortSettingsReport

Controls access to the Port Authority Report, which details port usage across the system.

%Ens_ProductionConfig

Controls access to production configuration activities.

%Ens_PurgeSchedule

Controls access to scheduling of InterSystems IRIS purge tasks.

%Ens_PubSub

Controls access to the Publish & Subscribe (or PubSub) pages in the Management Portal.

%Ens_PurgeSettings

Controls access to the Purge Management Data page in the Management Portal and controls the default settings for manually purging production-related data.

%Ens_Queues

Controls access to queue data.

Note: If you want to perform an activity related to an active message you will also need access to job data which uses the **%Ens_Jobs** resource.

%Ens_RestrictedUI_SystemDefaultSettings

Restricts a user to editing only the system default settings to which they have been given USE permission. For more information, see Security for System Default Settings.

%Ens_RecordMap

Controls access to interoperability record maps.

%Ens_RoutingRules

Controls access to routing rules.

%Ens_Rules

Controls access to all interoperability rules.

%Ens_SettingsReportConfig

Controls access to the Setting Report Configuration page, which enables you to specify the namespace that stores data about port usage.

%Ens_SystemDefaultConfig

Controls access to system–wide default settings.

%Ens_SystemDefaultSettings_AllowedIPAddresses

Allows user to manage the AllowedIPAddresses system default setting even when they are restricted from managing other system default settings. For more information, see Security for System Default Settings.

%Ens_SystemDefaultSettings_IPAddress

Allows user to manage the IPAddress system default setting even when they are restricted from managing other system default settings. For more information, see Security for System Default Settings.

%Ens_SystemDefaultSettings_Port

Allows user to manage the Port system default setting even when they are restricted from managing other system default settings. For more information, see Security for System Default Settings.

%Ens_SystemDefaultSettings_Server

Allows user to manage the Server system default setting even when they are restricted from managing other system default settings. For more information, see Security for System Default Settings.

%Ens_WorkflowConfig

Controls access to workflow roles and users.

Note: In many cases, the 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 roles and privileges use the less granular resource, but you can choose alternative roles with more selective privileges.

A.2.2.1 Security for System Default Settings

Assigning USE permission to the **%Ens_RestrictedUI_SystemDefaultSettings** resource restricts a user from creating, editing, or deleting system default settings for Interoperability productions. This restriction applies only to managing system default settings in the Management Portal and does not prevent administrators from editing the global directly.

You can grant exceptions to this general restriction by assigning USE privileges to the **%Ens_SystemDefaultSettings**_*setting* resource, where *setting* is the case-sensitive name of a setting. The system includes predefined resources for four settings:

- **%Ens_SystemDefaultSettings_AllowedIPAddresses** — Allows users to manage the AllowedIPAddresses setting from the Management Portal even when blocked from managing other system default settings.
- **%Ens_SystemDefaultSettings_IPAddress** — Allows users to manage the IPAddress setting from the Management Portal even when blocked from managing other system default settings.
- **%Ens_SystemDefaultSettings_Port** — Allows users to manage the Port setting from the Management Portal even when blocked from managing other system default settings.
- **%Ens_SystemDefaultSettings_Server** — Allows users to manage the Server setting from the Management Portal even when blocked from managing other system default settings.

For more information about system default settings, see Defining System Default Settings. For instructions on creating resources, see Create or Edit a Resource.

A.3 Predefined Roles Related to Productions

InterSystems IRIS also contains a set of predefined roles related to productions. Their names each begin with the %EnsRole_ prefix. These are roles designed to reasonably secure your InterSystems IRIS instances in both development and live environments. The following descriptions contain an overview of the perceived job responsibilities of members of the role and how these roles relate to other roles.

%EnsRole_Administrator

Role for a trusted and skilled 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 default system settings; and so on. This administrator has almost unlimited ability to control the InterSystems IRIS Interoperability environment, but cannot change code components other than to deploy updates.

This role is intentionally distinct from InterSystems IRIS administrative roles and does not grant the user any non-production 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 InterSystems IRIS 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. 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 InterSystems IRIS Interoperability developer role have full programming power and as such, can modify DTL, BPL, and record maps. InterSystems IRIS 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 InterSystems IRIS development tasks as well as the 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 Interoperability 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 InterSystems IRIS 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 InterSystems IRIS Interoperability 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 InterSystems IRIS Interoperability roles or create your own roles, assigning them permissions to the InterSystems IRIS resources. If you upgrade your InterSystems IRIS instance, the procedures reset the default 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 roles on the **System Administration** > **Security** > **Roles** page of the Management Portal.

These roles only cover functions in the Interoperability menus of the Management Portal. Users in your environment likely require additional InterSystems IRIS roles. For details, see Roles.

A.3.1 Roles for Business Rule Editor

The Business Rule Editor has an additional role, **%EnsRole_InteropEditorsAPI**, that is granted through the web application `/api/interop-editors`. This role does not need to be assigned to users, but will be temporarily applied when a user accesses the application. For more details on how roles and web applications work together, see Applications in the Authorization Guide. Details on the resources associated with this role are provided below:

- In order to access `/api/interop-editors` and obtain the **%EnsRole_InteropEditorsAPI**, a user must first have one of:
  - **%Ens_Rules:READ**

  - **%Ens_BusinessRules:READ**
  - **%Ens_RoutingRules:READ**

- **%EnsRole_InteropEditorsAPI** grants a user **%Development:USE**.
- **%EnsRole_InteropEditorsAPI** also grants a user EXECUTE privileges on the following SQL resources:
  - **%Ens_Config.Production_Extent**
  - **%Studio_SourceControl.Interface_MenuItems**
  - **%Studio_SourceControl.Interface_MainMenus**
  - **%Dictionary.ClassDefinition_SubclassOf**
  - **%Dictionary.ClassDefinition_Summary**
  - **%Atlier_v1_Utils.Extension_UserAction**
  - **%Atlier_v1_Utils.Extension_AfterUserAction**
  - **%Atlier_v1_Utils.Extension_ExtensionEnabled**
  - **%Atlier_v1_Utils.Extension_GetStatus**
  - **%Atlier_v1_Utils.Extension_GetMenus**

A.4 Default Privileges of the Predefined Roles

This section lists the default privileges that each role has for each 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 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		

Resource	%EnsRole _Administrator	%EnsRole _Developer*	%EnsRole _Monitor	%EnsRole _Operator
%Ens_MessageDiscard	Use	Use		
%Ens_MessageEditResend	Use	Use		
%Ens_MessageHeader	Use	Use		Use
%Ens_MessageResend	Use	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
%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				

Resource	%EnsRole _Administrator	%EnsRole _Developer	%EnsRole _Monitor	%EnsRole _Operator
%Ens_EDISchema	Read	Read, Write		
%Ens_JBH				
%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	Use	Read		Read
%Ens_PurgeSettings	Read, Write	Read, Write		
%Ens_Queues	Read, Write	Read, Write		Read
%Ens_RecordMap				
%Ens_RoutingRules				
%Ens_Rules*		Read, Write		
%Ens_SystemDefaultConfig	Read, Write	Read		Read
%Ens_WorkflowConfig	Write	Read, Write		Read

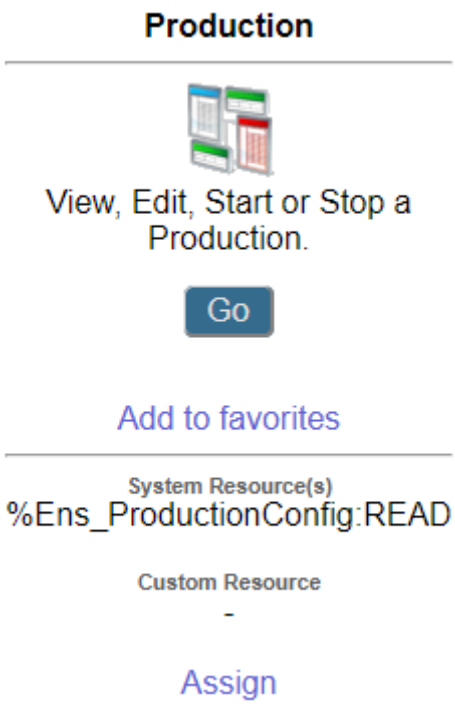
Additional roles have the following privileges

- The **%EnsRole_WebDeveloper** role has the same privileges as **%EnsRole_Developer**, except for access to the **%Ens_PurgeSettings** resource.
- 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 InterSystems IRIS. 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, if you select **Interoperability** > **Configure** and then click to the right of **Production** on 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 Use Custom Resources with the Management Portal.

A.5 Default SQL Privileges of the Predefined Roles

Several InterSystems IRIS Interoperability pages in the Management Portal use SQL queries to retrieve information; therefore, users must have privileges on the appropriate tables to view this information. This section shows how InterSystems IRIS 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 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 roles have SELECT privileges on a subset of the 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

SQL Table Name	%EnsRole _RulesDeveloper	%EnsRole _Monitor	%EnsRole _Operator
Ens_Util.Log		SELECT	SELECT
Ens_Util.Schedule			SELECT

InterSystems IRIS also grants privileges on two stored procedures:

- EXECUTE privileges on the **Ens_Config.Production_Extent** stored procedure (used by the system to list and load productions) to **`%EnsRole_Administrator`** and **`%EnsRole_Developer`**
- EXECUTE privileges on the **Ens.IsASub** stored procedure (used by the system 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 EXECUTE privileges on the **Ens.IsASub** to the role or user. To see if a specific role has this privilege in an interoperability-enabled 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: InterSystems IRIS 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 Security

For information on customizing security, see the following topics:

- Use Custom Resources with the Management Portal
- Manage User Accounts
- Web Applications

B

Finding Information on Menu Items

For reference, this page describes where to find information for the options of the **Interoperability** menu of the Management Portal.

Also see Controlling Access to Management Portal Functions.

For general information on the Management Portal, see the Management Portal Overview and the Management Portal Page Reference.

B.1 Configure Menu

Every item on this menu is described in the following topics:

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
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 System Default Settings
Purge Data Settings	View or edit default settings for purging production data in a given namespace	Configuring Default Settings for Manually Purging Production Data
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
Record Maps	View or edit file format record maps	Creating a Record Map
CSV Record Wizard	Create record maps from delimited files	Using the CSV Record Wizard

B.3 View Menu

Every item on this menu is described, 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
Interface Maps	View interface maps	Viewing Interface Maps
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
Data Transformations	View a list of data transformations	Developing DTL Transformations
Business Rules	View a list of business and routing rules	Business Rule List

Option	Purpose	See
Record Maps	View a list of record maps	Creating a Record Map
Productions	Manage other productions	Viewing the Production List

B.5 Monitor Menu

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

Option	Purpose	See
System Monitor	View the InterSystems IRIS® system monitor to monitor productions in all namespaces	Using the System Monitor
Production Monitor	Monitor a single production more closely	Monitoring a Production
Queues	View queues	Monitoring Production Queues
Jobs	View 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
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 **Interoperability** > **Interoperate** menu options provide access to tasks you perform to view and transform Electronic Data Interchange (EDI) messages to determine how you want your production to handle them. To display the page, click **Interoperate** in the 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	X12 Schemas and Available Tools
UN/EDIFACT	View, import, and remove EDIFACT schemas; view and transform EDIFACT documents	Available Tools for Working with EDIFACT
XML	View, import, export, and remove XML schemas; view and transform XML documents	Tools for Working with XML Documents and Schemas

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
Data Transformations	View the results of data transformations on sample messages	Developing DTL Transformations