
# First Look: Scaling Systems for User Volume with InterSystems Distributed Caching

Version 2018.1
2018-10-22

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

*First Look: Scaling Systems for User Volume with InterSystems Distributed Caching*
InterSystems IRIS Data Platform Version 2018.1 2018-10-22








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

# First Look: Scaling Systems for User Volume with InterSystems Distributed Caching

This First Look guide introduces you to how InterSystems IRIS Data Platform™ can scale for user volume by using application servers for distributed caching, leveraging the Enterprise Cache Protocol (ECP).

This guide presents an introduction to scaling with distributed caching architecture and walks through some initial tasks associated with deploying an InterSystems IRIS™ distributed cache cluster. Once you've completed this guide, you will have a basic understanding of how a distributed cache cluster works and how to set it up.

These activities are designed to use only the default settings and features, so that you can acquaint yourself with the fundamentals of the feature without having to deal with details (though these may be important when performing an implementation). For full documentation on using distributed caching and ECP with InterSystems IRIS, see the list of resources in the For More Information section at the end of this guide.

## 1 The Problem: Scaling for User Volume

When users connect to your InterSystems IRIS databases via applications, they need quick and efficient access to the data. Whether your enterprise is small, large, or in between, a high number of concurrent user requests against the databases — *user volume* — can cause performance problems on the system that hosts the databases. This can potentially affect many more users, making them wait longer to receive the information they need. And in a dynamic business, user volume can grow rapidly, further impacting performance.

In particular, if a lot of users are executing many different queries, the size of those queries can outgrow the cache, meaning that they can no longer be stored in memory and instead need to read data from the disk. This inefficient process causes bottlenecks and performance problems. You can increase the memory and cache size on the system (vertical scaling), but that solution can be expensive, inflexible, and ultimately limited by the maximum capabilities of the hardware. Spreading the user workload over multiple systems (horizontal scaling) is a more flexible, efficient, and scalable solution.

## 2 The Solution: Distributed Caching

To improve the speed and efficiency of users' access to the data, InterSystems IRIS can use distributed caching. This technology allows InterSystems IRIS to store the database cache on multiple *application servers*. User volume can then be distributed across those servers, thus increasing the cache efficiency. The internode communication that makes this possible is enabled by ECP, the Enterprise Cache Protocol.

Using distributed caching, you can enable users who make similar queries to share a portion of the cache, which is hosted on an application server clustered with the data server where your data is hosted. The actual data remains on the data server, but caches are maintained on the application servers for faster user access. The data server takes care of keeping the cached data up-to-date on each application server in the enterprise.

With a distributed cache cluster, you can easily scale your solution by adding or removing application servers as needed. All application servers automatically maintain their own connections to the data server, and attempt to recover the connection if it drops.

You can configure application servers and their associated data servers on the individual cluster instances using their Management Portals, or deploy and configure a cluster using InterSystems Cloud Manager (ICM). For more information on ICM, see First Look: ICM and the InterSystems Cloud Manager Guide.

# 3 How Does Distributed Caching Work?

When you deploy an InterSystems IRIS distributed cache cluster, you will designate one instance as the data server, and one or more instances as application servers. The instances do not need to run on the same operating system or hardware; they only need to conform to the InterSystems IRIS system requirements.

- The data server performs like a standard InterSystems IRIS server, hosting databases in namespaces and providing the data to other systems upon request.
- The application servers receive data requests from applications. When a user opens an application, instead of connecting to the data server, it connects to an application server. The user won't notice anything different. The application server fetches the necessary data from the data server and provides it to the user.
- The application server stores the data in its own cache, so that the next time any user requests the same data, the application server can provide it without needing to contact the data server again.
- The data server monitors all of the application servers to make sure that the data in their caches is up-to-date. The data server also handles data locks for the whole system.
- If the connection between an application server and the data server is lost, the application server automatically attempts to reconnect and recover any needed data.
- You can design your applications to direct users who make similar queries to the same application server. That way, the users can share a cache that includes the data they need the most. For example, in a healthcare setting, you might have clinicians running a particular set of queries and front-desk staff running different queries, using the same application and the same underlying data; those sets of users can be grouped together on separate application servers. As another example, if the cluster handles multiple applications, each application's users can be directed to their own application server(s) for maximum cache efficiency.

# 4 Trying Distributed Caching for Yourself

It’s easy to set up a distributed cache cluster with InterSystems IRIS. This simple procedure walks you through the basic steps of configuring ECP on several instances.

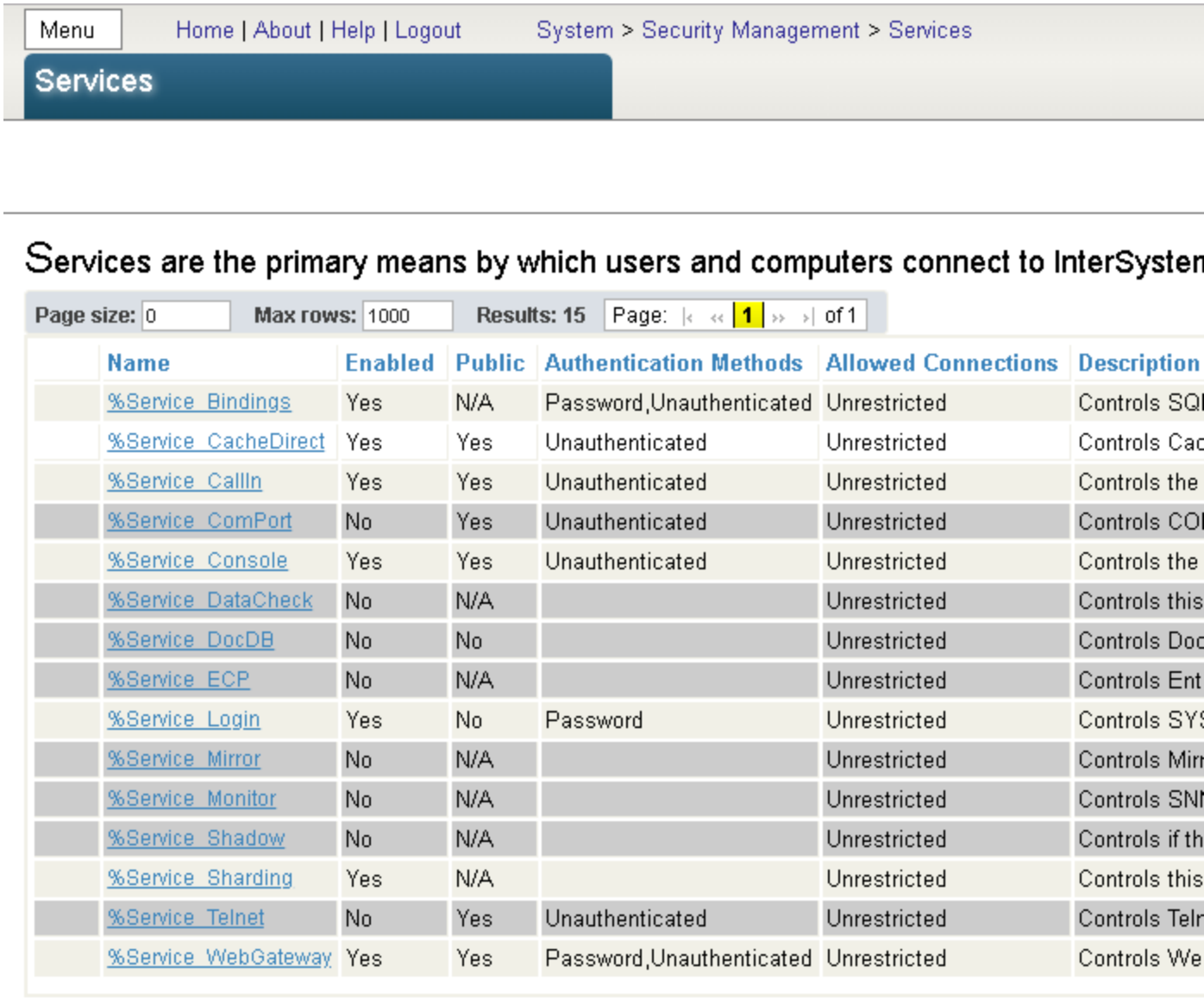
For the purposes of this example, we'll be setting up one InterSystems IRIS instance as a data server and two more instances as application servers. This means you will need to install (or have already available) three licensed instances. For a quick overview of the installation process, see Quick Start: InterSystems IRIS Installation.

**Note:** To give you a taste of distributed caching without bogging you down in details, we’ve kept this exploration simple; for example, we’ve had you use as many default settings as possible. When you bring this feature to your production systems, though, you may want to configure some settings (for example, security settings) differently. The sources provided at the end of this document will give you more details.

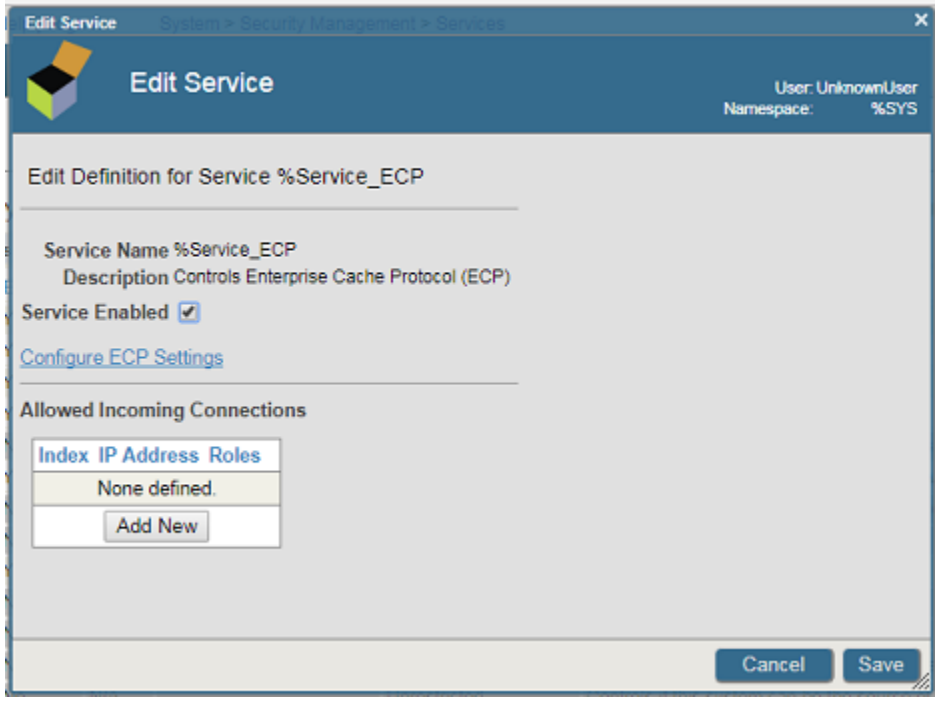
## 4.1 Enabling the ECP Service

First, enable the ECP service on all three instances as follows:

1. Log in to the Management Portal and go to the **Services** page (**System Administration** > **Security** > **Services**):




2. Select **%Service_ecp**. On the **Edit Service** page, select the **Service Enabled** check box, and then select **Save**.



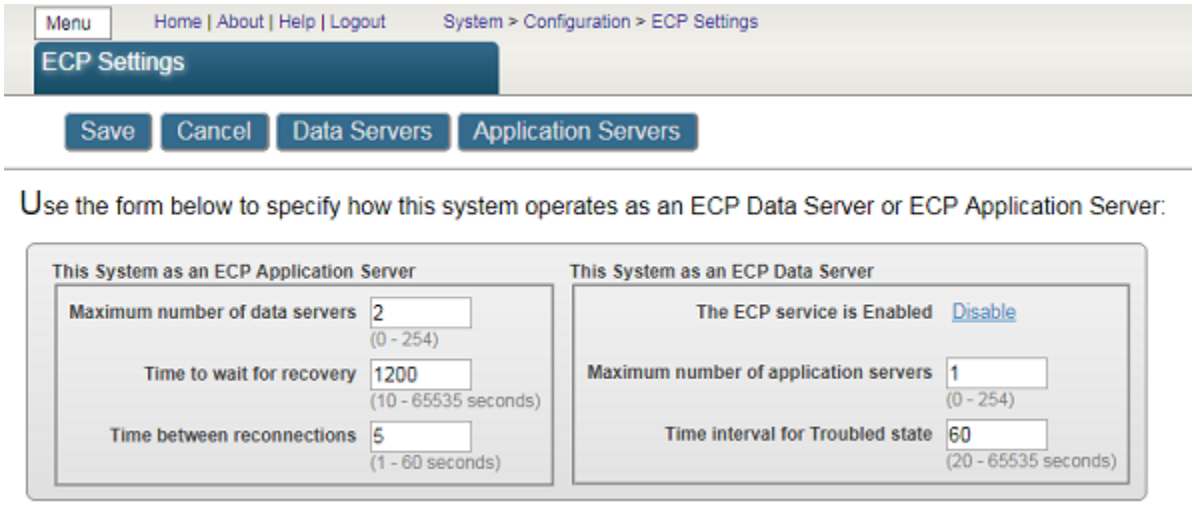

You have now enabled ECP on your system. There are just a few more steps to finish the setup for the data server and the two application servers.

## 4.2 Configuring the Data Server

On the system that will be your data server, there are just two more quick steps to finish the setup. First, you need to increase the number of allowed application servers from the default of one. Then, we'll create a new database for our application servers to connect to. Of course, in a production environment, you would already have a database in use.

To finish the data server configuration:

1. In the Management Portal, go to the **ECP Settings** page (**System Administration** > **Configuration** > **Connectivity** > **ECP Settings**):



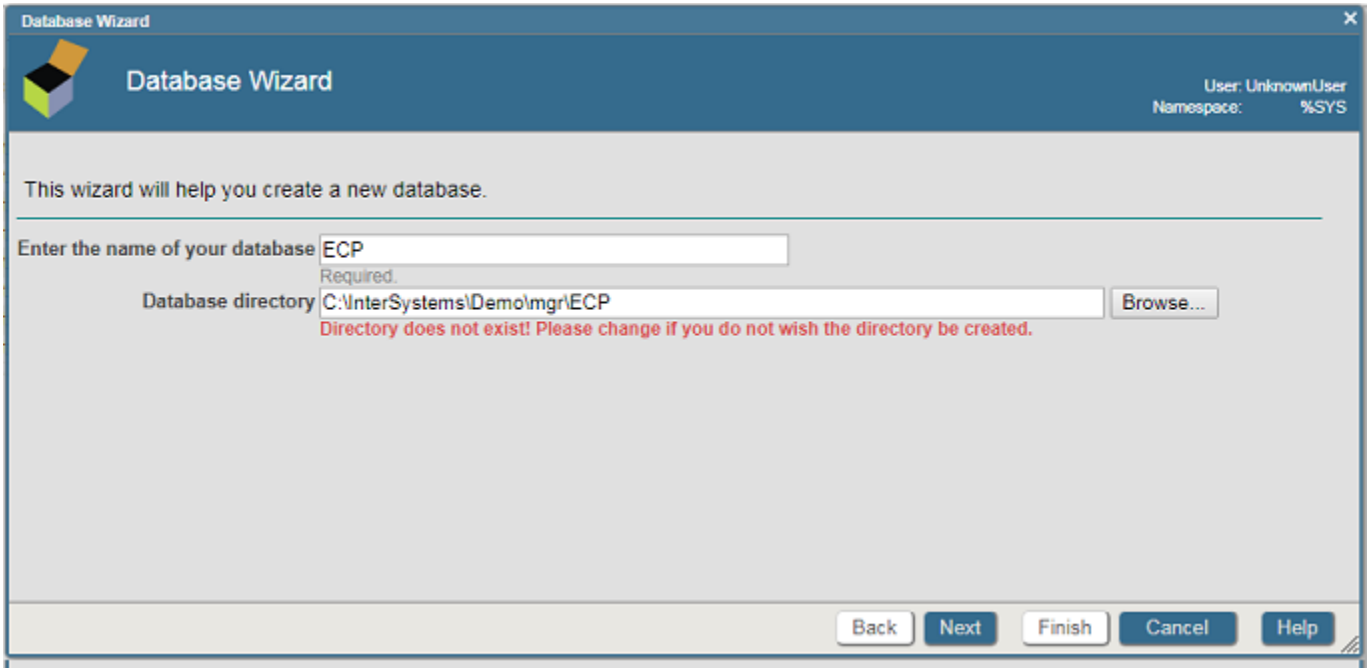

2. In the section labeled **This System as an ECP Data Server**, set the **Maximum number of application servers** to 2. Select **Save**.

3. Restart the instance.

For more details about creating a data server and setting the available options, see Preparing the Data Server in the "Horizontally Scaling Systems for User Volume with InterSystems Distributed Caching" chapter of the *Scalability Guide*.

To create a new database for this exercise:

1. In the Management Portal, go to the **Local Databases** page (**System Administration** > **Configuration** > **System Configuration** > **Local Databases**).
2. Select **Create New Database**.
3. Enter a name for the new database. For this exercise, we are calling it ECP.




   In the example shown above, our InterSystems IRIS instance is called `Demo`.

4. Select **Next** and then **Finish**.

You've created your new database, and your data server is ready to go.

In the next sections, you will set up the two application servers and configure them to be able to communicate with the data server. For this purpose, you will need to know the data server's superserver port number, so let's look that up now:

1. On the data server instance, in the Management Portal, go to the **Memory and Startup** page (**System Administration** > **Configuration** > **System Configuration** > **Memory and Startup**).
2. Make a note of the **Superserver Port Number**. you will need that for the procedures in the next section.

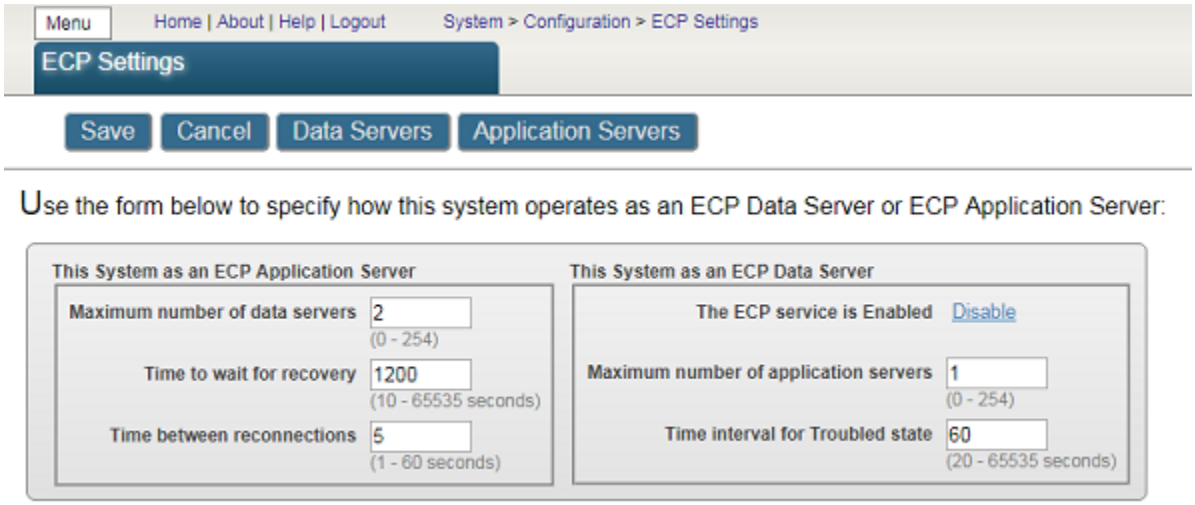
## 4.3 Configuring the Application Servers

Next, we'll set up our other two instances as application servers. We'll configure each application server to point to the data server, and we'll create a new namespace on each application server. The new namespaces will be mapped to the database we created on the data server.

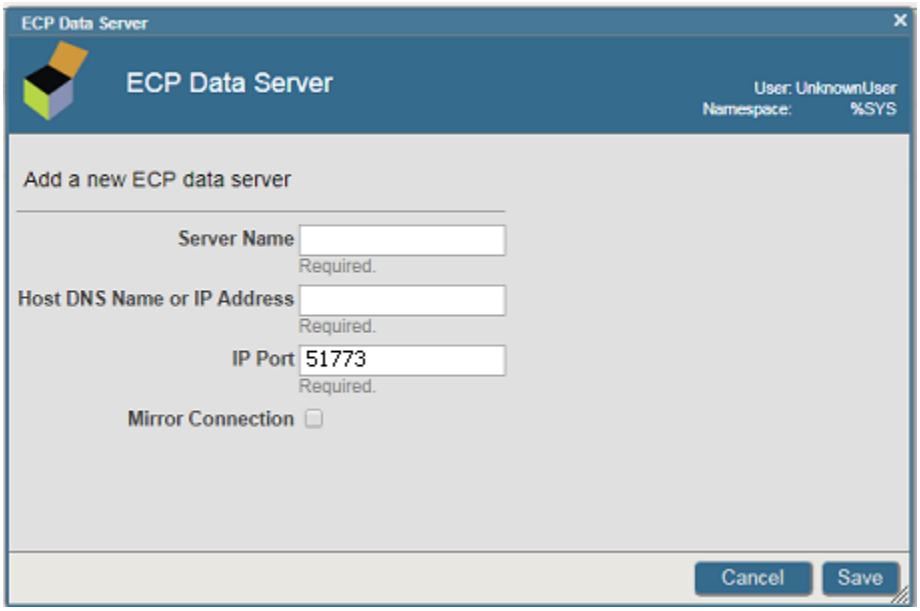
Perform the procedures in the next sections twice — once on each application server.

### 4.3.1 Setting Up the Application Server

1. Log in to the Management Portal and go to the **ECP Settings** page (**System Administration** > **Configuration** > **Connectivity** > **ECP Settings**):




2. Select **Data Servers** and then select **Add Server**.



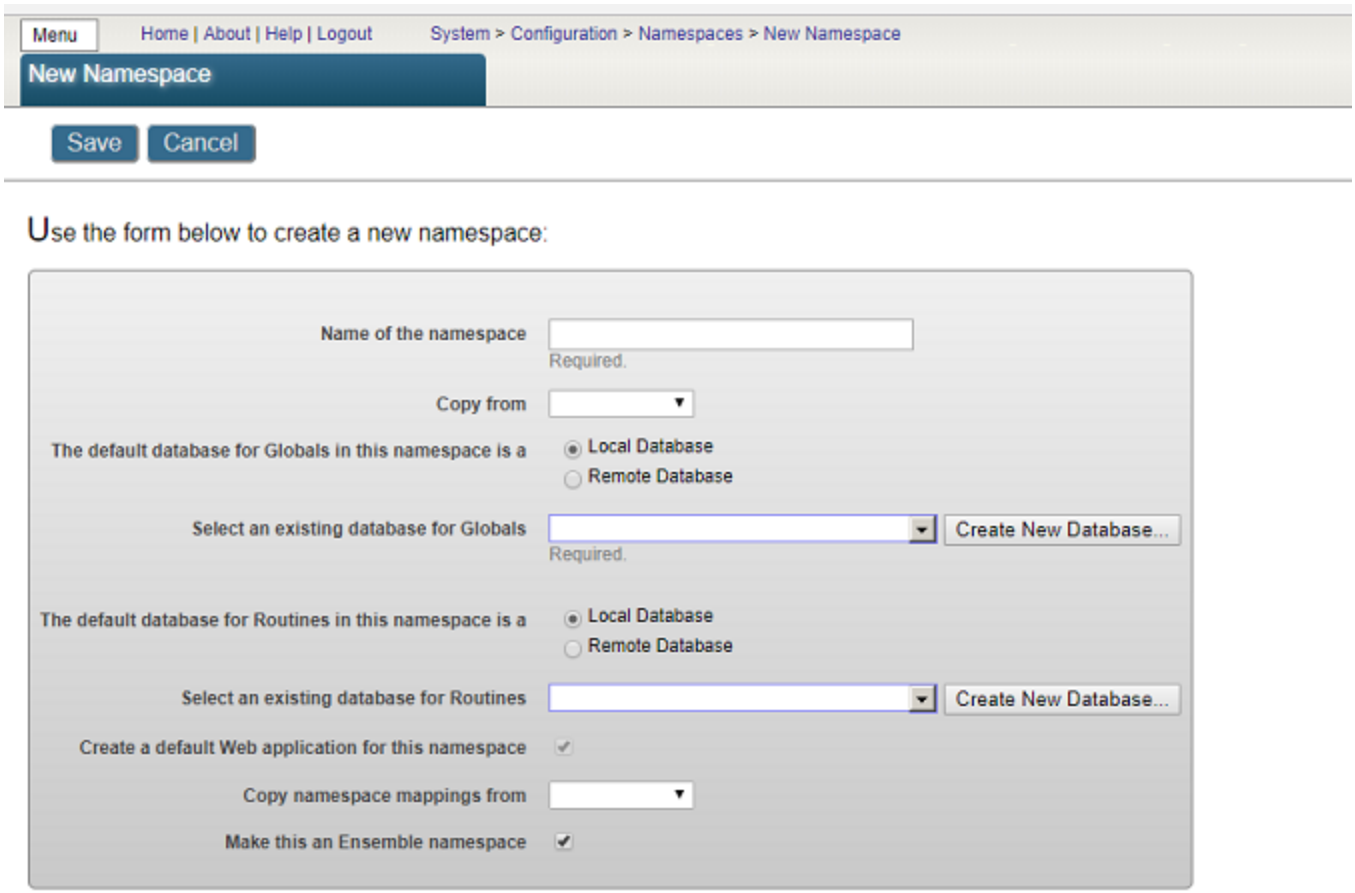

3. Fill in the required information:
   - **Server Name** — Enter a name or label to identify this server. It doesn't need to be the same as the instance name.
   - **Host DNS Name or IP Address** — Enter the location information for the data server you created in the previous section.
   - **IP Port** — Enter the superserver port number that you looked up on the data server in the previous section.
4. Select **Save**. Your new data server now appears on the list. It may take a few moments for the application server to connect to the data server and verify the connection.

### 4.3.2 Creating the Namespace and the Remote Database

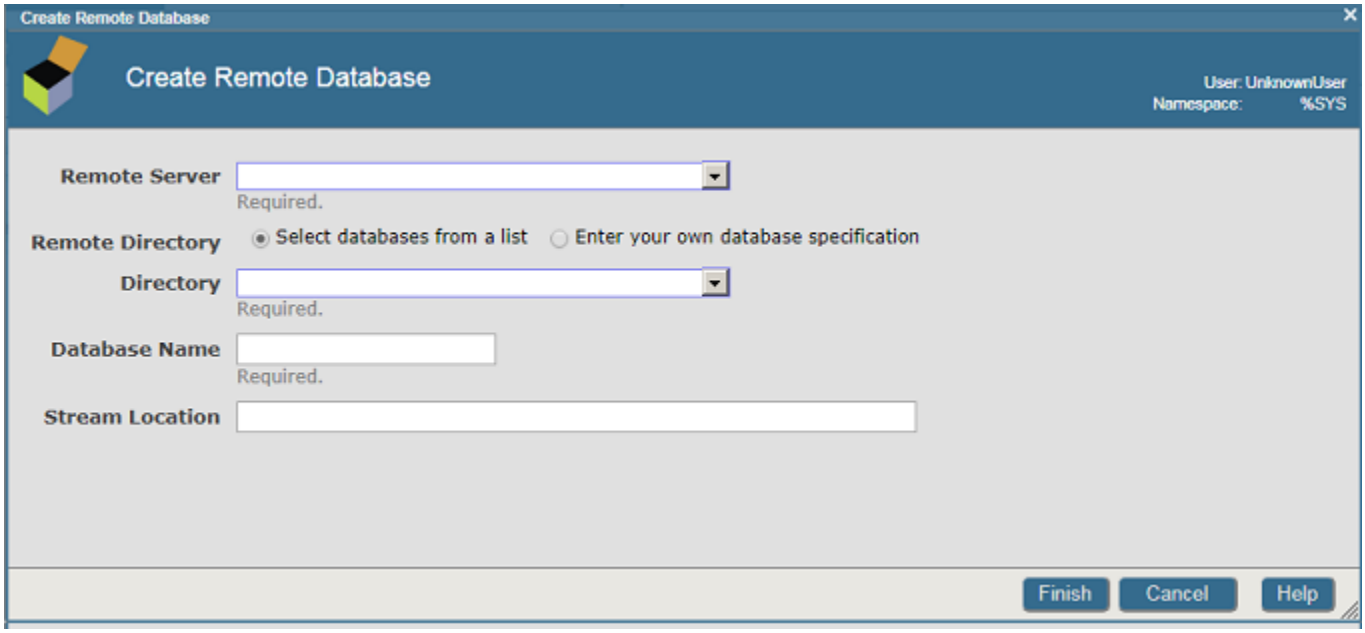
Now that you have connected your application servers to your data server, you need to create a namespace on each application server. This namespace will be local to the application server, but instead of containing a local database, it will be mapped to a remote database — that is, the ECP database on the data server, which we created in the previous section.

Remember, we are creating two application servers, so you should perform this procedure twice.

1. In the Management Portal, go to the **Namespaces** page (**System Administration** > **Configuration** > **System Configuration** > **Namespaces**).
2. Select **Create New Namespace**.




3. In the **Name of the namespace** field, enter ECPNS.
4. For **The default database for Globals in this namespace is a**, select **Remote Database**. Then select the **Create New Database...** button. This opens the **Create Remote Database** window.




5. Fill in the required information:
   - **Remote Server** — Use the drop-down menu to select the data server that we created in the previous section.
   - **Remote Directory** — Select the directory on the data server instance that contains the database.
   - **Database Name** — Enter a name for the database. This can be the same as its name on the data server, which is `ECP` in our example.
6. Select **Finish**. The window closes, and you're returned to the **New Namespace** page. You should see that the database you just created is now shown in the **Select an existing database for Globals** field.
7. For **The default database for Routines in this namespace is a**, select **Remote Database**. You should now be able to select the new database you just created from the drop-down menu.
8. Clear the **Make this an interoperability-enabled namespace** check box.
9. Select **Save**. The new namespace now appears on the list.

For more details about creating a namespace and its associated database, see "Create/Modify a Namespace" in the "Configuring InterSystems IRIS" chapter of the *System Administration Guide*. For background information, see "Namespaces and Databases" in the *Orientation Guide for Server-Side Programming*.

You're done! You've successfully created a cluster that has one data server and two application servers. In the next section, we'll test the connections to make sure that all three instances are communicating with each other correctly.

## 4.4 Testing the Setup

Now that we have enabled the ECP service and set up our two application servers with namespaces pointing to the database on the data server, it's time to do a quick test to make sure that the three systems are communicating with each other. To accomplish this, we'll set a simple global on one application server, then read and change it on the second application server.

To learn more about globals, see Using Globals.

1. On one application server, log in to the Terminal and change to the namespace that you created in the previous section. For our example, we called the namespace `ECPNS`, so we would do:

```
USER>zn "ECPNS"
ECPNS>
```

2. Create a global simply by giving it a value:

```
ECPNS> set ^MyGlobal = "My Value"
```

3. On the other application server, log in to the Terminal and change to the ECPNS namespace as described above.
4. Read the global:

```
ECPNS> write ^MyGlobal
My Value
```

   This shows that the two application servers are communicating properly with the data server. You used the application server to create the global, but because you were working in the namespace that contains the remote database, the global was actually created on the data server. That's why the other application server can read it. Of course, this is only an example, but the mechanism is the same, regardless of whether you're manually setting and then reading a global on the Terminal, or having a large number of users issuing thousands of transactions per second through a dozen application servers fronting the same data server. ECP will ensure that the data is kept in sync and guarantee transactional consistency for all those users' interactions with the system.

5. If you like, you can view the global on the data server instance as a final check. In the Management Portal, go to the Local Databases page (**System Administration** > **Configuration** > **System Configuration** > **Local Databases**). Locate the database that your application servers are pointing to, and select Globals for that database. You should see `MyGlobal` on the list.

# 5 Learn More About Distributed Caching and ECP

To learn more about using distributed caching and ECP with InterSystems IRIS, see the following resources:

- "Horizontally Scaling Systems for User Volume with InterSystems Distributed Caching" chapter of the *Scalability Guide*
- Sample Mirroring Architecture and Network Configurations, Redirecting Application Connections Following Failover or Disaster Recovery, Configuring Application Server Connections to a Mirror, and other distributed caching and ECP-related sections in the "Mirroring" chapter of the *High Availability Guide*