

Configure the source code viewer

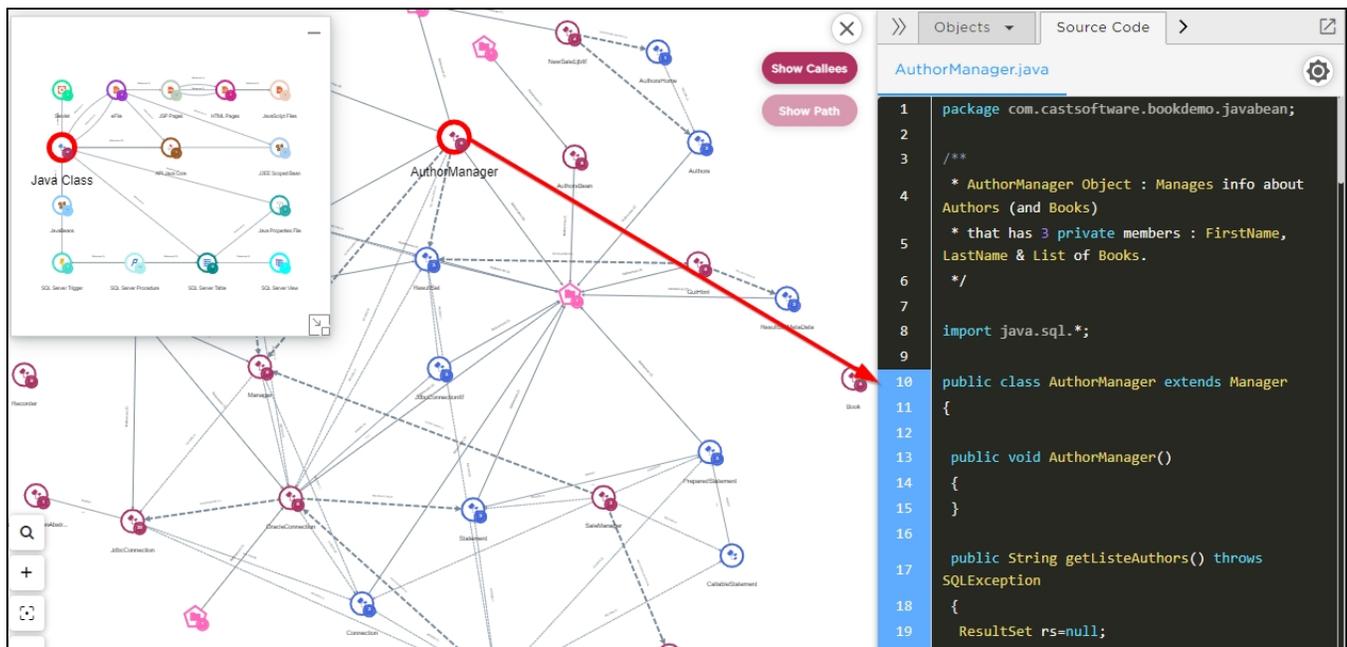
- [Introduction](#)
- [Step 1 - Identify instance names](#)
- [Step 2 - Configure the application.properties file](#)
 - [Examples](#)
 - [Single CAST Storage Service/PostgreSQL instance](#)
 - [Multiple CAST Storage Service/PostgreSQL instances](#)
- [Step 3 - apply the changes](#)

Summary: how to configure the source code viewer so that you can view source code of objects in CAST Imaging.

Introduction

It is possible to view object source code direct in CAST Imaging. Source code is available by right clicking objects in the Investigation view (available after a double clicking an item at Level 5) and selecting **Show source code**. Below is an explanation of how to configure the source code viewer.

Click to expand



The screenshot displays the CAST Imaging interface. On the left, a network diagram shows various objects connected by lines. A red circle highlights the 'AuthorManager' object. A red arrow points from this object to the source code viewer on the right. The source code viewer shows the following code:

```
1 package com.castsoftware.bookdemo.javabeans;
2
3 /**
4  * AuthorManager Object : Manages info about
5  * Authors (and Books)
6  * that has 3 private members : FirstName,
7  * LastName & List of Books.
8  */
9
10 import java.sql.*;
11
12 public class AuthorManager extends Manager
13 {
14     public void AuthorManager()
15     {
16     }
17
18     public String getListeAuthors() throws
19     SQLException
20     {
21         ResultSet rs=null;
```

Step 1 - Identify instance names

First identify the instance names for the **CAST Storage Service / PostgreSQL hosts** from which the Application data has been exported:

- Locate the Application **ZIPs** generated during the [export process](#) and unzip them
- Open each **.csv** file named **xxx_is_app.csv** with a text editor (where **xxx** is the Application name)
- Find the **instance name** on the line starting with **InstanceName** - in the example below the instance is an IP address **192.168.200.104**:

```
property,label,value
InstanceName,Instance Name,192.168.200.104
SchemaName,Schema Name,v8315_local
Extension,com.castsoftware.JEE-MavenHttp,2.0.6
```

- Make a note of all the instance names that you want to configure for source code viewing and match these to the CAST Storage Service /PostgreSQL instances.

i Note that the instance name will be the same as the "host" name used in the export batch scripts.

Step 2 - Configure the application.properties file

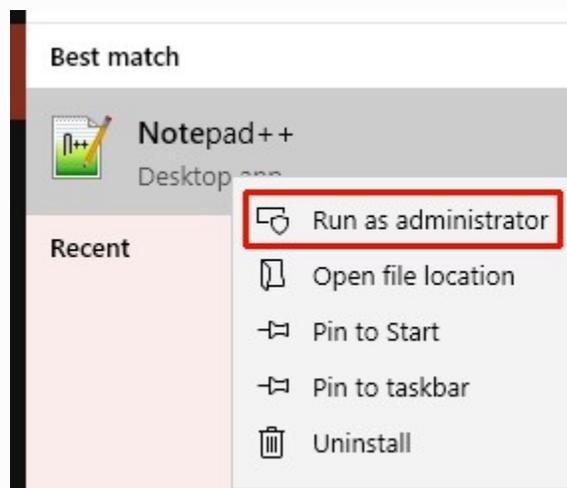
Locate and edit the following file:

```
Traditional Windows installer: %APPDATA%\CAST\ImagingSystem\sourcecode\application.properties
```

```
Docker Installer (located in the unzipped extension folder):  
server\sourcecode\application.properties or sourcecode\application.properties
```

i Microsoft Windows traditional installer

The file is located in the protected %APPDATA% location, therefore you must open the .properties file with **elevated permission** (this is usually achieved by right clicking your text editor in the Windows start menu and selecting **Run as administrator**):



Linux

You may need to use elevated permissions to edit this file (for example use **sudo**).

Now locate the following lines in the `application.properties` file and ensure they match the **CAST Storage Service / PostgreSQL** instances that host the Applications you have **generated and imported**. The `application.properties` file will be predefined with two "dummy" CAST Storage Service/PostgreSQL instances as shown below, these can be removed:

```
datasource.url=jdbc:postgresql://127.0.0.1:2284/postgres,jdbc:postgresql://127.0.0.1:2284/postgres  
datasource.username=operator,operator  
datasource.password=CastAIP,CastAIP  
datasource.alias=localhost,local
```

- `datasource.url` >>> The syntax is as follows: `jdbc:postgresql://<host or IP address>:<port>/postgres`. Multiple instances should be separated by a comma.
- `datasource.username` >>> For the vast majority of CAST Storage Service / PostgreSQL instances, this will be `operator`. Multiple usernames should be separated by a comma.
- `datasource.password` >>> For the vast majority of CAST Storage Service / PostgreSQL instances, this will be `CastAIP`. Multiple password should be separated by a comma.
- `datasource.alias` >>> This entry must match the "instance name" in the `xxx_is_app.csv` file - usually this matches the host name of the CAST Storage Service from which the Application originates. Separate with a comma for multiples instances.

Examples

Single CAST Storage Service/PostgreSQL instance

Single CAST Storage Service/PostgreSQL instance called `my_postgresql` with default username and password:

```
datasource.url=jdbc:postgresql://my_postgresql:2282/postgres
datasource.username=operator
datasource.password=CastAIP
datasource.alias=my_postgresql
```

Multiple CAST Storage Service/PostgreSQL instances

Multiple CAST Storage Service/PostgreSQL instances are defined using an "alias" system based on the instance name given to the CAST Storage Service / PostgreSQL host during the export. In this example we have one instance called **my_postgresql1** and another called **my_postgresql2** both with default username and password:

- ensure that the hosts are added in the same order on each line
- for the `datasource.alias` line, ensure that the instance name you discovered in the previous step is used in the correct position. I.e if you have an **alias** "my_postgresql1" in first position, then the details for the **url**, **username** and **password** lines for this host must also be in the first position on the other lines.

```
datasource.url=jdbc:postgresql://my_postgresql1:2282/postgres,jdbc:postgresql://my_postgresql2:2282/postgres
datasource.username=operator,operator
datasource.password=CastAIP,CASTAIP
datasource.alias=my_postgresql1,my_postgresql2
```



See also [Admin Center - Managing application aliases](#) for more information about alias management.

Step 3 - apply the changes

Save the file and ensure that you restart the **CAST Imaging System - sourcecode service** Windows service or the **sourcecode Docker** container in order for the new configuration to be taken into account.