

Using a custom CAST Storage Service or PostgreSQL database other than the default postgres

- [Introduction](#)
- [AIP Console](#)
 - [1.27](#)
 - [2.0](#)
- [CAST Dashboards and RestAPI](#)
- [Legacy tools in AIP Core](#)
 - [CAST Server Manager / CAST Enlighten](#)
 - [CAST Management Studio/pmx files](#)
 - [CAST Transaction Configuration Center](#)

Introduction

Changes have been made to AIP Core and other CAST AIP tools to make it possible to connect to a **custom** CAST Storage Service/PostgreSQL database other than the default "postgres" (previously custom databases were not permitted for use with CAST). The current compatibility matrix is as below:

	Supported?	Supported in	Notes
AIP Console		1.27.0-funcnel / 2.0.0-funcnel	When using AIP Console with automatic synchronization to CAST Imaging, custom databases are supported as long as the releases of AIP Console and CAST Imaging are supported for this scenario.
AIP Core		8.3.40	This includes any tool/application delivered with AIP Core such as CAST Server Manager , CAST Management Studio , CAST Enlighten , CAST Transaction Configuration Center , CSS Admin Tools etc.
Standalone CAST Dashboards and RestAPI		Any release.	-
CAST Imaging export tools		2.6.0-funcnel	This includes exporter.exe and etl-automation.exe .
CAST Sherlock		9.0.2-funcnel	9.0.2-funcnel is also embedded in AIP Core 8.3.41.
Architecture Checker (as an extension)		-	Not supported yet.

To be able to exploit this change, if required, please see information below.

AIP Console

Release **1.27 / 2.0** has this ability built in:

1.27

AIP Node installation:

lzPack - Installation of AIP Console

AIP Node configuration

Step 5 of 12

Server configuration:

Port:

AIP location:

CSS server configuration:

Database host and port:

Database username:

Database password:

Database name:

AIP Node app-node-app.properties:

```
# =====
# CSS Server parameters
# -----
database.server.name=192.168.200.104:2282
database.server.user=operator
# to encrypt the password use aip-encryption-tool
database.server.password=CRYPTED2:90B1A6EC1618661401B724DB5AC34595
database.name=postgres
# Enable SSL with client certificate
database.server.ssl=false
# Path to the INI file containing the certificate locations
database.server.ssl.iniPath=
# Minimum number of idle connections that are maintained in the conn
database.server.minimumIdle=10
# Maximum size that the connection pool is allowed to reach, includi
database.server.maximumPoolSize=10
```

Measurement schema configuration:

Measurement

SETTINGS

Schema name*
general_measure_127

Host UBUNTU2004	Port 2282
Username operator	Password
Database name* postgres	

When adding a new CAST Storage Service/PostgreSQL instance:

Add database connection

Host	Port
Username	Password
Database name postgres	

Use SSL

CANCEL **ADD**

CAST Dashboards and RestAPI

CAST Storage Service/PostgreSQL connections are defined using JDBC URLs which specify the database "postgres" out of the box:

```
WAR 1.x - context.xml
<Resource name="jdbc/domains/AED" url="jdbc:postgresql://localhost:2280/postgres"

WAR/ZIP 2.x - application.properties
restapi.datasource[0].url=jdbc:postgresql://localhost:2282/postgres
spring.datasource.url=jdbc:postgresql://localhost:2282/postgres?ApplicationName=DASHBOARDS
ntSchema=cast_dashboards
```

These JDBC URLs must be modified to change the "postgres" database to the custom database name (as described in <https://jdbc.postgresql.org/documentation/head/connect.html>) for example:

```
WAR 1.x - context.xml
<Resource name="jdbc/domains/AED" url="jdbc:postgresql://localhost:2280/custom_database"

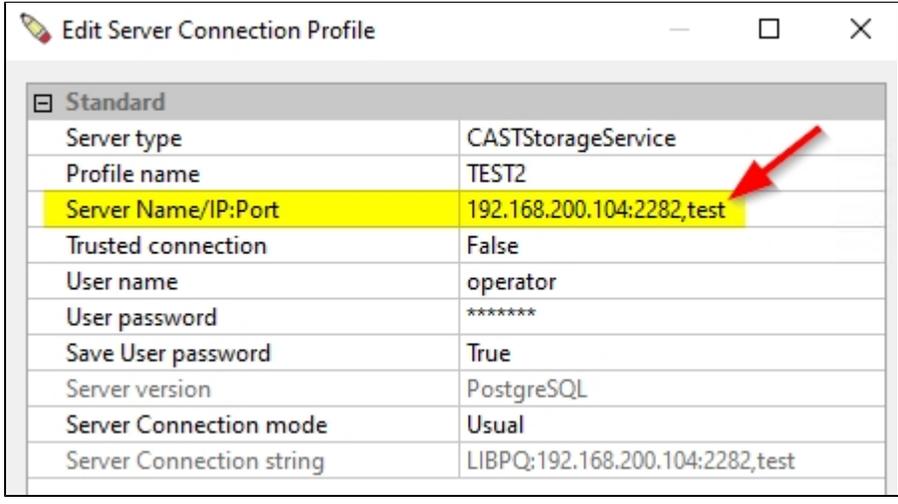
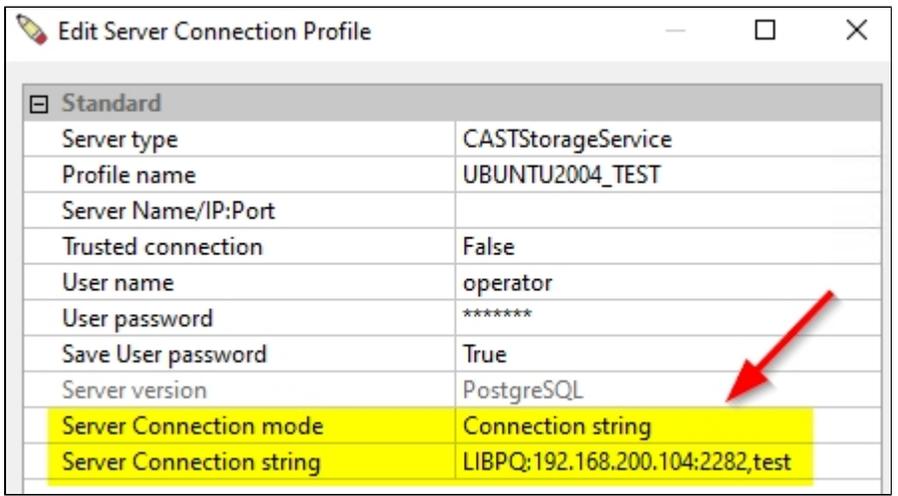
WAR/ZIP 2.x - application.properties
restapi.datasource[0].url=jdbc:postgresql://localhost:2282/custom_database
spring.datasource.url=jdbc:postgresql://localhost:2282/custom_database?ApplicationName=DASHBOARDS
ntSchema=cast_dashboards
```

 See [Standalone CAST Dashboard deployment process](#) for more information.

Legacy tools in AIP Core

CAST Server Manager / CAST Enlighten

The UI has not been modified, however to connect to a database other than "postgres", create a new profile and either use the **Server name/IP:port** field or the **Connection string** field, for example:

<p>Standard connection</p>	<p><host/IP_address>: <port>,<database_name></p>	 <table border="1"> <thead> <tr> <th colspan="2">Standard</th> </tr> </thead> <tbody> <tr><td>Server type</td><td>CASTStorageService</td></tr> <tr><td>Profile name</td><td>TEST2</td></tr> <tr><td>Server Name/IP:Port</td><td>192.168.200.104:2282,test</td></tr> <tr><td>Trusted connection</td><td>False</td></tr> <tr><td>User name</td><td>operator</td></tr> <tr><td>User password</td><td>*****</td></tr> <tr><td>Save User password</td><td>True</td></tr> <tr><td>Server version</td><td>PostgreSQL</td></tr> <tr><td>Server Connection mode</td><td>Usual</td></tr> <tr><td>Server Connection string</td><td>LIBPQ:192.168.200.104:2282,test</td></tr> </tbody> </table>	Standard		Server type	CASTStorageService	Profile name	TEST2	Server Name/IP:Port	192.168.200.104:2282,test	Trusted connection	False	User name	operator	User password	*****	Save User password	True	Server version	PostgreSQL	Server Connection mode	Usual	Server Connection string	LIBPQ:192.168.200.104:2282,test
Standard																								
Server type	CASTStorageService																							
Profile name	TEST2																							
Server Name/IP:Port	192.168.200.104:2282,test																							
Trusted connection	False																							
User name	operator																							
User password	*****																							
Save User password	True																							
Server version	PostgreSQL																							
Server Connection mode	Usual																							
Server Connection string	LIBPQ:192.168.200.104:2282,test																							
<p>Connection string</p>	<p>LIBPQ:<host /IP_address>:<port>, <database_name></p>	 <table border="1"> <thead> <tr> <th colspan="2">Standard</th> </tr> </thead> <tbody> <tr><td>Server type</td><td>CASTStorageService</td></tr> <tr><td>Profile name</td><td>UBUNTU2004_TEST</td></tr> <tr><td>Server Name/IP:Port</td><td></td></tr> <tr><td>Trusted connection</td><td>False</td></tr> <tr><td>User name</td><td>operator</td></tr> <tr><td>User password</td><td>*****</td></tr> <tr><td>Save User password</td><td>True</td></tr> <tr><td>Server version</td><td>PostgreSQL</td></tr> <tr><td>Server Connection mode</td><td>Connection string</td></tr> <tr><td>Server Connection string</td><td>LIBPQ:192.168.200.104:2282,test</td></tr> </tbody> </table>	Standard		Server type	CASTStorageService	Profile name	UBUNTU2004_TEST	Server Name/IP:Port		Trusted connection	False	User name	operator	User password	*****	Save User password	True	Server version	PostgreSQL	Server Connection mode	Connection string	Server Connection string	LIBPQ:192.168.200.104:2282,test
Standard																								
Server type	CASTStorageService																							
Profile name	UBUNTU2004_TEST																							
Server Name/IP:Port																								
Trusted connection	False																							
User name	operator																							
User password	*****																							
Save User password	True																							
Server version	PostgreSQL																							
Server Connection mode	Connection string																							
Server Connection string	LIBPQ:192.168.200.104:2282,test																							



- See [SRV - Creating Connection Profiles](#) and [ENL - Creating Connection Profiles](#).
- When you install a set of Application schemas with **CAST Server Manager**, a connection profile (PMX format) for use with CAST Management Studio will be automatically created for your target CAST Storage Service/PostgreSQL instance, including the custom database name.

CAST Management Studio/pmx files

The UI for tools using .pmx connection files (such as CAST Management Studio) has not been modified to allow a connection to a database other than "postgres", however, it is possible to build/edit your own custom **cast-ms.connectionProfiles.pmx** and specify your custom database name in this .pmx file. See [Using the CAST-Connection-Profile-Configuration.py file](#).

The **Database Connections Manager** in CAST Management Studio will show the details about a connection to a custom database:

CAST-MS Database connections Manager

Name	Server details	Container
v8340_mngt on CastStorageService_192.168.200.104:2282,my_customdb	//LIBPQ:192.168.200.104:2282,my_customdb:2282/postgres	v8340_mngt

Import... Export...

CAST Transaction Configuration Center

The UI has not been modified, however to connect to a database other than "postgres", use the Quick Connection option and specify the custom database name (or select a PMX based profile if one has already been created):

Quick Connection

Connection ✕

CSS Oracle SQL Server

Server address
192.168.200.104

Port
2282

Database
my_customdb

User name
operator

Password

Schema (management base)
v8340_mngt

Ok Cancel

**PMX
based
profile**

CAST-MS Database connections manager

Select a connection profile:

Name	Server details	Container
v8340_mngt on CastStorageService_192.168.200.104.2282.my_customdb	//LIBPQ:192.168.200.104.2282.my_customdb.2282	v8340_mngt
v8338_mngt on CastStorageService_192.168.200.104.2282	//192.168.200.104.2282	v8338_mngt
test_mngt on CastStorageService_192.168.200.104.2282.test	//LIBPQ:192.168.200.104.2282.test.2282	test_mngt
sdm_folder_mngt on CastStorageService_UBUNTU2004:2282	//UBUNTU2004:2282	sdm_folder_mngt
My CAST Storage Service profile	//localhost:2280	
meudonv2_mngt on CastStorageService_UBUNTU2004:2282	//UBUNTU2004:2282	meudonv2_mngt
meudon_v2_mngt on CastStorageService_192.168.200.121.2285	//192.168.200.121:2285	meudon_v2_mngt
meudon_v2_mngt on CastStorageService_192.168.200.107.2285	//192.168.200.107:2285	meudon_v2_mngt
meudon_mngt on CastStorageService_UBUNTU2004:2282	//UBUNTU2004:2282	meudon_mngt

Quick Connection Connect Cancel

 See [TCC - CAST Transaction Configuration Center - Connection](#).