

Deploying a Service

Especially in fusion teams, developers want to make a finished feature or an intermediate state of a service available to other team members for testing while they continue to work on the same service. This is easily possible in the Designer by using the test environment for developing (refer to [Working with the Test Environment](#) for details) and deploying to the test server only when necessary. You only need to deploy your service if you want to allow other team members to test new features or if you want to run regression tests against your test server.



We recommend using container deployment as the default deployment target starting with PAS 23.1. For further information see:

- [Designer Guide > Deployment as Container](#)
- [Administration Guide > Controlling Containerized xUML Services](#)

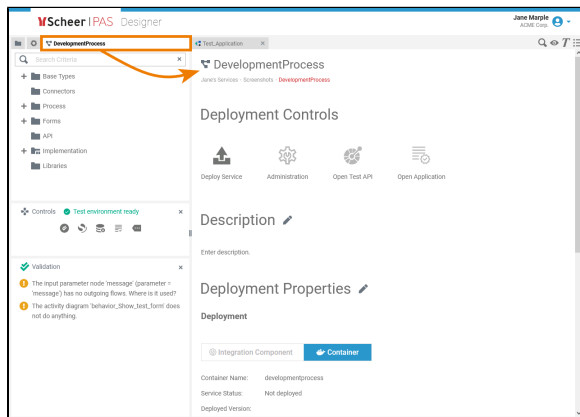
On this Page:

- [Executing the Deployed Service](#)
 - [Example: Test Service vs. Deployed Service](#)

Related Pages:

- [Validating and Testing a Service](#)
- [Running Designer Applications](#)
- [Using the Deployment Controls](#)
- [Deployment to the Integration Component \(Bridge\)](#)
- [Deployment as Container](#)

The **Deployment Controls** are available on the service details page (see chapter [Managing the Service Details](#) for further information). Click on the **Service** panel tab (service name) to open the details page:



Deployment Properties

Deployment



Integration Component



Container

Container Name: developmentprocess

Select your deployment target the first time you deploy.



Prerequisites

- **Container deployment:** Your user must be assigned the **xuml_container_admin** profile.
- **Integration (Bridge) deployment:** Your user must be assigned the **integration_user** profile and he must have an integration user account created by an integration administrator.

Refer to [Administration Guide > Overview of Standard Profiles](#).

Enable edit mode  of the deployment properties and choose between **Integration Component** or **Container**.

Deployment Properties

Deployment



Integration Component



Container

Container Name: developmentprocess

Service Status: Not deployed

Deployed Version:

General

Version: 1.0.0

Category: Development

Enable Validation: ☒

Enable Angular Build: ☒

We recommend to enter a version number and increase the number with each deployment.



Expert Advice

By default, each newly created service gets number **0.1.0** assigned. We recommend to increase the version number before redeploying each time you have made relevant changes to the service. You can change the version number in section **General** of the **Deployment Properties**. Follow the concept of [semantic versioning](#).

In case of deployment problems, comparing the version number and the number of the deployed version can help to find out which version of the service is running.

Save  the deployment properties to continue.

Deployment Controls



Deploy Service



Administration



Open Test API



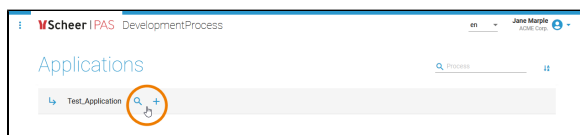
Open Application

Now use the deployment controls to deploy the service to your chosen target:

Icon	Description
	Use this option to deploy the service to the selected deployment target.
	Use this option to open the management UI of the selected deployment target: PAS Administration for Container Integration for Integration Component
	Use this option to open the test UI for the service-related API.
	Use this option to open the deployed service in a new browser tab.

Executing the Deployed Service

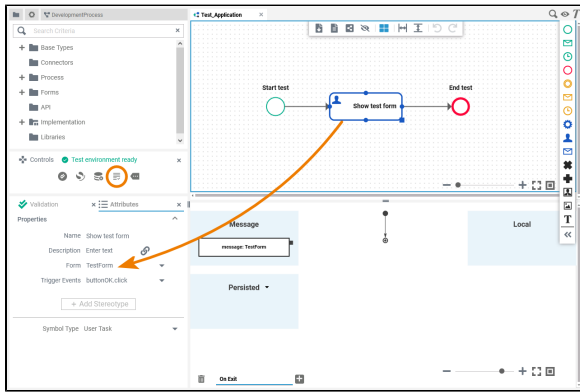
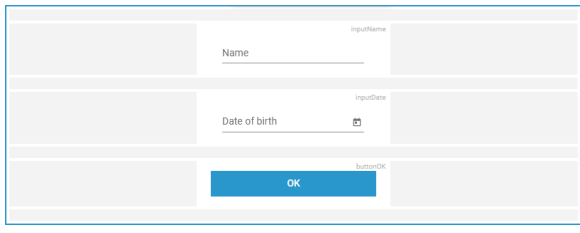
The deployed service is accessible via option **Open Application** in the deployment controls section. The start page of the deployed service looks exactly the same as that of the test service:



Applications Test_Application + Start test	The handling of the deployed service is also the same: • Users can create new instances and execute the process. • Users can access the instance overview.
--	---

The difference is: The deployed version does not change, even if the developer changes the service in the test environment.

Example: Test Service vs. Deployed Service

	A developer created a simple test application, containing only a user task with an assigned test form.
	The test form contains two input fields and a button.
Deployment Controls Deploy Service Administration Open Test API Open Application	The developer deploys this version.

First Name inputFirstName		After the deployment of this version, he starts to extend the test form.
Last Name inputLastName		
Date of birth inputDate		
Upload Your Profile Image profileImage Drag and drop here or 📁 Browse		
Continue buttonContinue		

Test Service

✖ Controls

✔ Test environment ready

✕

✔

🔄

📁

☰

👤

⋮

Deployed Service

Deployment Controls

📁

Deploy Service

⚙️

Administration

🔗

Open Test API

☰

👤

Open Application

er
o
p
e
n
s
t
h
e
t
e
s
t
s
e
r
v
i
c
e.

Applications

↳ Test_Application



Start test



Applications

↳ Test_Application



Start test



t
h
e
p
r
o
c
e
s
s
t
o
c
r
e
a
t
e
a
n
e
w
i
n
s
t
a
n

Name

Date of birth



OK

First Name

Last Name

Date of birth



Upload Your Profile Image

Drag and drop file here
or



Browse

Continue

T
h
e
a
d
a
p
t
e
d
f
o
r
m
o
p
e
n
s.