



Use the OpenEdge Command Center

Copyright

Visit the following page online to see Progress Software Corporation's current Product Documentation Copyright Notice/Trademark Legend: <https://www.progress.com/legal/documentation-copyright>.

October 2026

Product version: Progress OpenEdge Command Center 2.1

Updated: 2026/10/12

Table of Contents

Preface.....	9
Learn about OpenEdge Command Center.....	11
OpenEdge Command Center components.....	12
OpenEdge Command Center console dashboard.....	14
Authorization server.....	16
Roles in Authorization server.....	16
Learn about external identity provider (IdP) authentication.....	17
Required IdP app roles for OpenEdge Command Center	18
How to.....	21
Manage services for OpenEdge Command Center server.....	23
Manage services for OpenEdge Command Center agent.....	24
Change logging level.....	25
Log in to OpenEdge Command Center.....	26
Switch authentication modes.....	28
Switch from local authentication to IdP authentication.....	29
Return to local authentication.....	30
Change or reset passwords.....	31
Configure email settings.....	32
Create a new user.....	33
Edit user details.....	34
Manage OpenEdge Command Center agents.....	35
Add agent label.....	35
Update agent name and label.....	36
Search for and filter agents.....	36
Unregister an agent.....	37
Assign agent permissions in IdP authentication mode.....	37
Manage OpenEdge databases.....	38
Add a new OpenEdge database connection	39
Start or stop OpenEdge databases.....	41
Edit OpenEdge database connections.....	41
Remove OpenEdge database connections.....	42
View schema, users, and roles of a selected database.....	43
Configure and manage PAS for OpenEdge instances.....	44
Configure PAS for OpenEdge instances.....	44
Manage PAS for OpenEdge instances.....	57
Manage ABL applications, web applications, and REST services.....	61
View ABL applications across multiple PAS for OpenEdge instances.....	62

Manage ABL applications.....	63
Manage web applications.....	65
Manage REST services.....	67
Edit ABL application.....	68
Create users and manage roles using Authorization server.....	73
Log in to Authorization server.....	74
Create a user.....	75
Assign roles to a user.....	75
Remove roles from a user.....	78
Create and manage custom roles.....	80
Frequently Asked Questions.....	80
Set up TLS for OpenEdge Command Center and MongoDB communication.....	82
Monitor OpenEdge resources using the OpenEdge Command Center agent.....	84
Set up OpenTelemetry Collector.....	85
OpenTelemetry metrics for OpenEdge database.....	87
Enable OpenEdge Command Center agent to collect performance metrics of OpenEdge database.....	89
Sample performance metrics data for OpenEdge database.....	91
OpenTelemetry metrics for PAS for OpenEdge.....	93
Enable OpenEdge Command Center agent to collect performance metrics of PAS for OpenEdge.....	97
Sample performance metrics data for PAS for OpenEdge.....	99
Performance impact and resilience of collecting performance metrics.....	102
Reset super admin user details.....	103
Troubleshoot identity provider authentication.....	104

Learn about the OpenEdge Command Center MCP Server 107

Learn how the OpenEdge Command Center MCP Server works.....	108
Explore tools and prompts.....	108
Understand the OpenEdge Command Center MCP Server prerequisites.....	114
Learn about authentication and authorization.....	114
Configure and manage the OpenEdge Command Center MCP Server.....	115
Start, stop, and check the OpenEdge Command Center MCP Server.....	117
Verify the OpenEdge Command Center MCP Server.....	118
Connect to the OpenEdge Command Center MCP Server	119
Create a PAT and connect an MCP client with it.....	119
Connect an MCP client with OpenID Connect (OIDC).....	120
Troubleshoot OpenEdge Command Center MCP Server connections.....	122

OpenEdge Command Center utilities.....125

OECCAGENT utility.....	125
OECCSERVER utility.....	126
RESETOECCDBPASSWORD utility.....	126

RESETSUPERADMIN utility.....	127
SWITCHAUTHMODE utility	129
OpenEdge Command Center performance results.....	133

Preface

Purpose

This manual is an introduction to the OpenEdge Command Center. It describes the installation and configuration procedures of the OpenEdge Command Center. It also describes the tasks performed by an administrator to monitor and manage OpenEdge environments using the OpenEdge Command Center.

Audience

This manual is designed as a guide and reference for OpenEdge Administrator and technical personnel responsible for installing and configuring OpenEdge Command Center.

Organization

- [Learn about OpenEdge Command Center](#) on page 11
This section provides an introduction to OpenEdge Command Center and its benefits to an OpenEdge administrator.
- [How to](#) on page 21
This section provides information about the various tasks that are required to operate the OpenEdge Command Center.
- [Learn about the OpenEdge Command Center MCP Server](#) on page 107
This section provides information about the OpenEdge Command Center MCP Server, how it works, and how to connect supported MCP clients.
- [OpenEdge Command Center utilities](#) on page 125
This section provides information about the OpenEdge Command Center utilities in alphabetical order.
- [OpenEdge Command Center performance results](#) on page 133
This section provides information about the performance results of OpenEdge Command Center under various scenarios

Documentation conventions

See [Documentation Conventions](#) for an explanation of the terminology, format, and typographical conventions used throughout the OpenEdge content library.

Learn about OpenEdge Command Center

Progress® OpenEdge® Command Center is a cloud-ready OpenEdge management console capable of managing multiple OpenEdge resources and versions across local and remote systems. The vision of OpenEdge Command Center is to offer substantial productivity gains for the day-to-day administration and management of the OpenEdge platform while providing flexibility for today's environment where system resources are volatile.

OpenEdge Command Center supports local authentication and external identity provider (IdP) authentication, but you can activate only one authentication mode at a time. When IdP authentication is active, the identity provider must include the user's assigned OpenEdge Command Center app roles in the access token. OpenEdge Command Center uses these roles to determine the user's access. For more information, see [Learn about external identity provider \(IdP\) authentication](#).

The OpenEdge Command Center Server installer includes the OpenEdge Command Center MCP Server. The MCP Server lets you connect MCP-compatible clients, such as GitHub Copilot in Visual Studio Code, and retrieve information about the OpenEdge Command Center resources that you are authorized to view. For details, see [Learn about the OpenEdge Command Center MCP Server](#).

OpenEdge Command Center is a complimentary tool that supports customers using OpenEdge 12.2 or later releases. For more information about tool life cycle policy, see *OpenEdge Tool Life Cycle* in the [OpenEdge Life Cycle Guide](#). You can install OpenEdge Command Center by using separate installers available on Progress Software Download Center.

To know more about installing OpenEdge Command Center, see "Install and configure OpenEdge Command Center server" and "Install and configure OpenEdge Command Center agent", see *Use the OpenEdge Command Center*.

OpenEdge Command Center features the latest UI and UX technologies. Currently, OpenEdge Command Center is built to effectively manage PAS for OpenEdge instances and OpenEdge databases. It aims to incrementally support everything you need to manage PAS for OpenEdge instances and databases. It simplifies the administrative overhead for system administrators by introducing lightweight agents and a centralized server, and it is designed to be highly available and extremely reliable.

OpenEdge Command Center uses open standard APIs to rapidly serve your integration use cases and provide timely information in a modern and easy-to-use interface. Intuitively designed, OpenEdge Command Center can be accessed with modern desktop or tablet-based web browsers. All the connections to the OpenEdge Command Center are secured using HTTPS, regardless of your platform.

To support your custom business needs for application integration and access, OpenEdge Command Center uses OpenAPI v3.0 REST APIs accompanied by interactive swagger documentation to help you add your own resources along the way and automate your tasks.

With the OpenEdge Command Center labeling feature, you easily manage all touch points of your PAS for OpenEdge components development pipeline from development to testing, staging, and production.

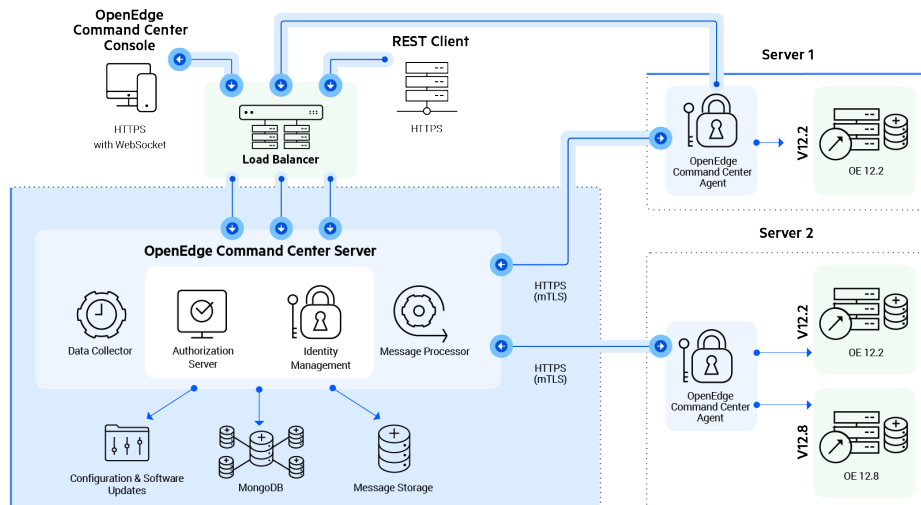
OpenEdge Command Center is composed of the OpenEdge Command Center console, the OpenEdge Command Center servers, and the OpenEdge Command Center agents.

For details, see the following topics:

- [OpenEdge Command Center components](#)
- [OpenEdge Command Center console dashboard](#)
- [Authorization server](#)
- [Learn about external identity provider \(IdP\) authentication](#)

OpenEdge Command Center components

Built for high availability, the OpenEdge Command Center server can be deployed on multiple servers, with the traffic distributed by a load balancer. Its built-in user management provides System Administrators and Application Administrators secure control of OpenEdge Command Center user permissions across all server instances. For complete resource visibility, OpenEdge Command Center offers automated component discovery for the OpenEdge version 12.2 and later releases, for all the PAS for OpenEdge deployments and OpenEdge databases.



The key components of OpenEdge Command Center are:

OpenEdge Command Center server

The OpenEdge Command Center server powers the OpenEdge Command Center console and APIs. You can deploy the OpenEdge Command Center server as a single node or, if configured for high availability, as a multi-node cluster of OpenEdge Command Center servers. For high availability, the server works with all load balancers that support HTTPS and WebSocket (WSS) protocols, such as Nginx, AWS, ELB/ALB, and more.

Authorization server

The Authorization server manages user access to the components of the OpenEdge Command Center, such as the OpenEdge Command Center server, OpenEdge Command Center agent, Authorization server, and their resources. It provides a framework for defining and enforcing authorization policies. For more information, see [Authorization server](#) on page 16.

OpenEdge Command Center data store

The OpenEdge Command Center uses MongoDB as its internal configuration database that stores all the OpenEdge Command Center configurations and details of discovered OpenEdge components. When the OpenEdge Command Center server is configured for high availability, multiple server instances of OpenEdge Command Center can be linked to the same database. In event of a planned or unplanned outage of an OpenEdge Command Center server, other OpenEdge Command Center servers can balance the load without impacting the end user.

To use high availability mode, MongoDB is installed on your first node system on which you install OpenEdge Command Center server. For the subsequent nodes, you point to the data folder of the first node, meaning the MongoDB configuration is shared across all the OpenEdge Command Center servers.

Note: The terms *OpenEdge Command Center data store*, *OpenEdge Command Center database* and *MongoDB* are used interchangeably in this guide and refer to the same database component.

OpenEdge Command Center agent

An OpenEdge Command Center agent is a lightweight process is co-located with your OpenEdge installation of the resources to manage. The agent has a one-to-many relationship with all your OpenEdge resources across (OpenEdge release 12.2 and later) installations on that system. Regardless of whether you have one or more OpenEdge installations on a server or host machine, you need only one installation of the OpenEdge Command Center agent. The agent is always backward compatible and can manage all versions of the OpenEdge release 12.2 and later. You can also use the OpenEdge Command Center agent to collect performance metrics of OpenEdge resources, such as OpenEdge database and PAS for OpenEdge using OpenTelemetry.

The agent initiates all the communications to an OpenEdge Command Center server, to ensure secure and protected access to the OpenEdge resources. The communication between them is through a secure, mutual TLS connection.

During the installation of the agent, you can configure the agent with the connection details of the server. The OpenEdge Command Center agent runs autonomously, monitoring OpenEdge components on its system. You need to ensure that the agent is up and running to manage OpenEdge resources on the remote system.

Browser clients

The OpenEdge Command Center server supports the HTTPS transport protocol to facilitate a secure and reliable connection for modern desktop, tablet, or iPad web browsers. Through your web browser, you can configure and manage OpenEdge resources using the OpenEdge Command Center console.

OpenEdge Command Center console dashboard

The OpenEdge Command Center console dashboard helps you to monitor and manage OpenEdge resources, including PAS for OpenEdge instances and OpenEdge databases. It provides an intuitive interface that includes a Getting Started video, banner notifications, and navigation bar to access the console pages. It also provides a list of newly auto-discovered OpenEdge Command Center agents, PAS for OpenEdge instances, and OpenEdge databases, and links to REST API documentation and user help resources. For more information, see [OpenEdge Command Center console dashboard](#) on page 14.

OpenEdge Command Center console dashboard

After you log in to OpenEdge Command Center, the console **dashboard** is displayed. The dashboard is the home page and consists of the following components to help you manage the PAS for OpenEdge instances and the OpenEdge databases:

Getting Started video

Watch the [Get started with the OpenEdge Command Center](#) video to get a high-level view of the product, to learn about the interface and its PAS for OpenEdge and OpenEdge database management features, and to learn where and how to perform the set of first-time tasks you need to complete to start using it.

Banner notification

The top right area of the console dashboard displays banner notification, alerting you to configure email settings. As a super administrator, the first task to complete is configuring your email settings. After that is done, you can create new users, manage your OpenEdge Command Center deployments, and receive email notifications sent by OpenEdge Command Center. After you configure your email settings, this banner notification is no longer displayed.

Discovered Resources

After you connect one or more OpenEdge Command Center agents, you can view these agents in the **Discovered Resources** panel. These agents list OpenEdge resources, such as PAS for OpenEdge instances and running OpenEdge databases, on their respective systems. As these resources are made discoverable by the agent, you can view these resources in the **Discovered Resources** panel.

If you click an agent in this panel, the **OECC Agents** page appears. For information about managing the agent from this page, see [Manage OpenEdge Command Center agents](#) on page 35.

Similarly, if you click an OpenEdge database or PAS for OpenEdge instance resource in this panel, the corresponding resource page appears. For information about managing the OpenEdge database, see [Manage OpenEdge databases](#) on page 38. For information about managing the PAS for OpenEdge instance, see [Configure and manage PAS for OpenEdge instances](#) on page 44.

Quick links

The **Quick Links** panel provides instant access to:

- OpenEdge Command Center online help, which is posted on the Progress Content Portal
- Progress community help
- Public REST APIs for business application use cases
- Version and build number of OpenEdge Command Center

Navigation bar

Running down the left side of the console is the **navigation bar**, which you can use for quick access to the following console pages:

Click . . .	. . . to display
	The OpenEdge Command Center console dashboard.
	The OECC Agents page, from which you can manage agents across all of your OpenEdge deployments. Additionally, you can monitor the status of your agents (running or stopped), as well as manage the labels associated with specific agents. For more information about agents, see Manage services for OpenEdge Command Center agent on page 24.
	The OpenEdge Databases page, which lists all the connected OpenEdge databases, including the automatically discovered running databases and manually added database connections. You can view the OpenEdge database details, such as name, labels, and status. From this page, you can add a new OpenEdge database connection, edit or remove existing OpenEdge database connections, and start or stop OpenEdge databases. For more information, see Manage OpenEdge databases on page 38.
	The PAS for OpenEdge page, which lists all instances that are available from an agent. You can view the instance details, such as its name, labels, and status. From this page, you can create, view, start, stop, clone, and delete PAS for OpenEdge instances. You can also modify the configuration instances, monitor CPU and memory consumption by PAS for OpenEdge instance, deploy and undeploy ABL and web applications for a selected PAS for OpenEdge instance. For more information, see Configure and manage PAS for OpenEdge instances on page 44.
	The ABL Applications page, from which you can view and manage ABL applications, web applications, and REST services. For more information, see Manage ABL applications, web applications, and REST services on page 61.
	The users page, from which you can create and manage users and user information, such as name, email, and role for OpenEdge Command Center servers. For more information, see Create a new user on page 33.
	The OECC system settings page, from which you can manage general settings for OpenEdge Command Center server. From this tab, you can manage OpenEdge Command Center database settings, update settings, OpenEdge Command Center agent settings, general settings, and repository settings.
	The email settings page, from which you can manage email notification and configuration settings. You can set the mail server host name, port, and authentication credentials. For more information, see Configure email settings on page 32.

Authorization server

The Authorization server is a component within the OpenEdge Command Center server. It is responsible for managing user access to systems, such as the OpenEdge Command Center server, the OpenEdge Command Center agent, and the Authorization server and their resources. It provides a framework for defining and enforcing authorization policies using Role-Based Access Control (RBAC). With RBAC, your access to the systems and their resources is determined by your assigned role. Each role comes with a predefined set of permissions that define the actions you are authorized to perform. For information about roles, see [Roles in Authorization server](#) on page 16.

The Authorization server starts automatically upon the initial startup of the OpenEdge Command Center server. Upon startup, a set of predefined roles is provisioned within the Authorization server. These roles enforce access control for the OpenEdge Command Center server and Authorization server.

Access control for the OpenEdge Command Center agent is enforced through its bootstrap policy. For more information about the bootstrap policy, see [Bootstrap policy file](#).

For more information, see "Bootstrap policy file" in *Install the OpenEdge Command Center*.

The Authorization server supports role-based access control at both the agent level and resource level. Agent-level roles provide access to all resources that the agent manages. Custom roles provide access to specific resources, such as a PAS for OpenEdge instance or OpenEdge database managed by the OpenEdge Command Center agent. You can assign custom roles to users to grant the least-privilege access.

The Authorization server provides user and role management capabilities. You can create users, assign roles, remove roles, and create and manage custom roles in the Authorization server through Authorization server REST APIs. For more information, see [Create users and manage roles using Authorization server](#) on page 73.

Roles in Authorization server

The Authorization server manages access control using predefined roles and policy-based roles. Each role is associated with a specific set of permissions that define the actions that a user can perform within a system.

The following table provides a brief description of predefined roles and policy-based roles in the Authorization server:

Role	Type	Description
OECC_ADMIN	Predefined	Grants access to the OpenEdge Command Center server.
AUTHZ_ADMIN	Predefined	Grants access to create users and manage roles.
AGENT_ADMIN	Predefined	Grants access to manage the AGENT_ADMIN and AGENT_RESOURCE_USER roles in the Authorization server for the OpenEdge Command Center agent.
AGENT_RESOURCE_USER	Policy-based	Grants access to all the OpenEdge resources that the OpenEdge Command Center agent manages.

In addition to predefined and policy-based roles, the Authorization server supports custom roles. Custom roles grant access to specific OpenEdge resources, such as a PAS for OpenEdge instance or OpenEdge database managed by an OpenEdge Command Center agent.

Role URN

A role Universal Resource Name (URN) identifies the roles within the Authorization server, indicating what a specific role has access to and where it is used. The Role URN format includes the resource name it represents, role name, and agent partition IDs (only when assigning roles to a user for an agent).

The following table lists roles and their corresponding URNs:

Role	Role URN
OECC_ADMIN	role:oecc/oecc_admin
AUTHZ_ADMIN	role:authz/authz_admin
AGENT_ADMIN	role:agent/<partitionid>/agent_admin
AGENT_RESOURCE_USER	role:agent/<partitionid>/agent_resource_user
<custom-role-name>	role:agent/<partitionid>/<custom-role-name>

Note: The agent partition ID is a unique identifier assigned to each agent. You can obtain the agent partition ID using the Retrieve specific agent details API on the OpenEdge Command Center server. For more information, see "Retrieve specific agent details" in *OpenEdge Command Center REST API Reference*.

Learn about external identity provider (IdP) authentication

OpenEdge Command Center supports local authentication and external identity provider (IdP) authentication, but only one authentication mode can be active at a time.

With local authentication, OpenEdge Command Center stores user credentials and manages user roles. In local authentication mode:

- OpenEdge Command Center creates and manages local user accounts.
- Users sign in with a local username and password.
- You manage users and assign roles in OpenEdge Command Center.

With IdP authentication, an OpenID Connect (OIDC)-compliant identity provider authenticates users and assigns roles.

IdP authentication can provides single sign-on (SSO), centralized user management and multi-factor authentication (MFA) managed by the organization's identity provider.

When IdP authentication is active, users sign in through the configured identity provider instead of entering their OpenEdge Command Center credentials.

In IdP authentication mode:

- The identity provider Administrator creates and manages users in the identity provider portal.

- Users sign in through the identity provider.
- The identity provider Administrator assigns app roles to users. The identity provider must include the assigned app roles in the `access_token`.
- The **Users** page and user-management REST endpoints are unavailable.
- OpenEdge Command Center uses the roles assigned by the identity provider to determine user access.
- OpenEdge Command Center continues to maintain the role definitions and permissions for those roles.
- The identity provider must provide the required OIDC scopes: `openid`, `profile`, `email`, and `offline_access`.
- A user without an email address cannot sign in. A user without the `role:oecc/oecc_admin` role can sign in but cannot access OpenEdge Command Center details.

OpenEdge Command Center supports OIDC-compliant identity providers such as Microsoft Entra ID and Okta configured with a custom authorization server.

To configure IdP authentication, an OpenEdge Command Center administrator must:

1. Configure the OpenEdge Command Center application in the identity provider portal.
2. Create the required app roles and assign them to users.
3. Ensure that the identity provider issues a JWT access token that contains the assigned app roles.
4. Configure the identity provider in OpenEdge Command Center by using the `SWITCHAUTHMODE` command-line utility.
5. Activate the configuration and restart the OpenEdge Command Center server. In high-availability deployments, restart each OpenEdge Command Center server so that all instances read the updated authentication configuration at startup.

You can use the `SWITCHAUTHMODE` utility on the OpenEdge Command Center server host to:

- Store and activate authentication configurations.
- Switch between authentication providers.
- Return to local authentication when required.

You can store multiple identity provider configurations, but only one authentication configuration can be active. Inactive configurations remain stored and can be activated later without entering their values again.

A role change done by the identity provider Administrator takes effect when the user signs in again or when OpenEdge Command Center refreshes the user session or UI.

Required IdP app roles for OpenEdge Command Center

Before users can be authorized in IdP authentication mode, the identity provider Administrator must create app roles, or equivalent role or group mappings, whose values match the OpenEdge Command Center role values exactly.

Fixed roles

Create the following roles in the identity provider:

Display name	Value	Description	Assign to
OECC Admin	role:oecc/oecc_admin	Full OpenEdge Command Center administration	Users or groups
AuthZ Admin	role:authz/authz_admin	Authorization management	Users or groups

Agent roles

OpenEdge Command Center creates agent roles automatically when an agent connects. Their values follow this pattern:

```
role:agent/<partitionId>/agent_admin  
role:agent/<partitionId>/agent_resource_user
```

Where <partitionId> is the partition ID of the agent—you copy it from the **Partition ID** column of the **Agents** grid. You must create a matching app role in the identity provider for each agent that the identity provider users must manage, and then assign users or groups to that role.

Role matching rules

Role values must exactly match the OpenEdge Command Center role values. Remember the following details when creating app roles:

- A role value that does not match any OpenEdge Command Center role is ignored, and grants no permissions.
- A role for an agent that no longer exists is ignored.
- A user who is assigned no OpenEdge Command Center app role can sign in but cannot view OpenEdge Command Center or agent details.
- A user who is assigned only an agent role can manage the resources of that agent only.

How to

Using OpenEdge Command Center, you can perform the following tasks:

For information how to . . .	. . . see
Manage services for OpenEdge Command Center	• Manage services for OpenEdge Command Center server on page 23 • Manage services for OpenEdge Command Center agent on page 24
Change the logging level	Change logging level on page 25
Log in	Log in to OpenEdge Command Center on page 26
Set your password	Change or reset passwords on page 31
Configure email settings	Configure email settings on page 32
Mange users	• Create a new user on page 33 • Edit user details on page 34
Manage agents	• Add agent label on page 35 • Update agent name and label on page 36 • Search for and filter agents on page 36 • Unregister an agent on page 37
Manage OpenEdge databases	• Add a new OpenEdge database connection on page 39

For information how to . . .	. . . see
	• Start or stop OpenEdge databases on page 41 • Edit OpenEdge database connections on page 41 • Remove OpenEdge database connections on page 42 • View schema, users, and roles of a selected database on page 43
Configure PAS for OpenEdge instances	• Configure PAS for OpenEdge instances on page 44 • Manage ABL applications or ABL web applications on page 53
Manage PAS for OpenEdge instances	• Create a PAS for OpenEdge instance on page 57 • Start or stop a PAS for OpenEdge instance on page 60 • Delete a PAS for OpenEdge instance on page 60 • Obtain process details on page 61
Manage ABL applications, web applications, and REST services	• View ABL applications across multiple PAS for OpenEdge instances on page 62 • Manage ABL applications on page 63 • Manage web applications on page 65 • Manage REST services on page 67 • Edit ABL application on page 68
Create users and manage roles using Authorization server	• Log in to Authorization server on page 74 • Create a user on page 75 • Assign roles to a user on page 75 • Remove roles from a user on page 78 • Frequently Asked Questions on page 80
Monitor OpenEdge resources using the OpenEdge Command Center agent	• Set up OpenTelemetry Collector on page 85 • OpenTelemetry metrics for OpenEdge database on page 87 • Enable OpenEdge Command Center agent to collect performance metrics of OpenEdge database on page 89 • Sample performance metrics data for OpenEdge database on page 91 • OpenTelemetry metrics for PAS for OpenEdge on page 93

For information how to . . .	. . . see
	• Enable OpenEdge Command Center agent to collect performance metrics of PAS for OpenEdge on page 97 • Sample performance metrics data for PAS for OpenEdge on page 99 • Performance impact and resilience of collecting performance metrics on page 102

For details, see the following topics:

- [Manage services for OpenEdge Command Center server](#)
- [Manage services for OpenEdge Command Center agent](#)
- [Change logging level](#)
- [Log in to OpenEdge Command Center](#)
- [Switch authentication modes](#)
- [Change or reset passwords](#)
- [Configure email settings](#)
- [Create a new user](#)
- [Edit user details](#)
- [Manage OpenEdge Command Center agents](#)
- [Manage OpenEdge databases](#)
- [Configure and manage PAS for OpenEdge instances](#)
- [Manage ABL applications, web applications, and REST services](#)
- [Create users and manage roles using Authorization server](#)
- [Set up TLS for OpenEdge Command Center and MongoDB communication](#)
- [Monitor OpenEdge resources using the OpenEdge Command Center agent](#)
- [Reset super admin user details](#)
- [Troubleshoot identity provider authentication](#)

Manage services for OpenEdge Command Center server

You can manage the OpenEdge Command Center server service by starting or stopping it as needed on both the Windows and Linux platforms as a user with administrator privileges. These actions are typically performed to apply configuration changes or troubleshoot issues and require administrator or superuser privileges.

Manage services for OpenEdge Command Center server on Windows

To start or stop the server installed as a service, perform the following steps:

1. Open **Task Manager** on the system where the server is installed using an account with administrator privileges.
2. Go to the **Services** tab and locate ProgressOpenEdgeCommandCenterServer2.1.
3. Right click the service name and perform the required action:
 - To start the service, select **Start**.
 - To stop the service, select **Stop**.

Manage services for OpenEdge Command Center server on Linux

To start or stop the server installed as a service, perform the following steps:

1. Open a terminal with superuser (root) privileges.
2. Run the appropriate command:
 - To start the service:

```
systemctl start ProgressOpenEdgeCommandCenterServer2.1
.service
```

- To stop the service:

```
systemctl stop ProgressOpenEdgeCommandCenterServer2.1
.service
```

Manage services for OpenEdge Command Center agent

After you start an OpenEdge Command Center agent, the status of the agent is updated in the **OECC Agents** page. Note that the status of an agent cannot be updated from the OpenEdge Command Center. You can start or stop the agent as a service. These actions are typically performed to apply configuration changes or troubleshoot issues and require administrator or superuser privileges.

Note: Do not use the Proenv environment command shell to start the OpenEdge Command Center agent. It can result in errors.

Manage services for OpenEdge Command Center agent on Windows

To start or stop the agent installed as a service on Windows, perform the following steps:

1. Open **Task Manager** on the system where the agent is installed using an account with administrator privileges.
2. Go to the **Services** tab and locate ProgressOpenEdgeCommandCenterAgent2.1.
3. Right click the service name and perform the required action:
 - To start the service, select **Start**.
 - To stop the service, select **Stop**.

Manage services for OpenEdge Command Center agent on UNIX or AIX

To start or stop the agent installed as a service, perform the following steps:

1. Open a terminal with superuser (root) privileges.
2. Run the appropriate command:

- On Linux or UNIX:

- To start the service:

```
systemctl start ProgressOpenEdgeCommandCenterAgent2.1.service
```

- To stop the service:

```
systemctl stop ProgressOpenEdgeCommandCenterAgent2.1.service
```

- On AIX:

- To start the service:

```
startsrc -e "OECC_AGENT=\"$OECC_AGENT\"" -s ProgressOECCAgent2.1
```

- To stop the service:

```
stopsrc -s ProgressOECCAgent2.1
```

Change logging level

Log files contain important information about the OpenEdge Command Center server and agents processes running on systems. The information in the log files is especially useful when you debug some issues, such as if an OpenEdge Command Center agent appears offline or as not running.

You can change the logging level to adjust the verbosity of information added to the log files. By default, the logging level for the OpenEdge Command Center server and agent logs is set as `info`. Based on your requirements, you can change the logging level to `debug`, `warn`, or `trace`.

Change logging level for the OpenEdge Command Center server logs

To change logging level for the OpenEdge Command Center server log, complete the following steps:

1. Navigate to the `/data/conf` folder in the OpenEdge Command Center installation directory.
2. Open the `system-config.json` file in a text editor.
3. Change the value of `logLevel` to the required level.
4. Save the changes and close the file.
5. Restart the OpenEdge Command Center server.

Change logging level for the OpenEdge Command Center agent logs

To change logging level for the OpenEdge Command Center agent logs, complete the following steps:

1. Navigate to the `resources` folder in the OpenEdge Command Center agent installation directory.
2. Open the `log4j2.properties` file in a text editor.
3. Change the value of `logger.app.level` to the required level.
4. Save the changes and close the file.
5. Restart the OpenEdge Command Center agent.

Log in to OpenEdge Command Center

This topic provides information on user accounts created on the successful completion of the OpenEdge Command Center server and MongoDB installation. It also describes how to access the OpenEdge Command Center dashboard and outlines the steps required to log in using the default super admin user account.

Default super admin and built-in accounts

After a successful installation of the OpenEdge Command Center server and MongoDB, the following accounts are created automatically:

- A default super admin user account for the initial login to the OpenEdge Command Center dashboard.
- A built-in user account for internal use in the OpenEdge Command Center database.

Default super admin account

The default super admin credentials for initial access to the OpenEdge Command Center dashboard are as follows:

- Username: admin
- Password: admin

Note:

- Progress recommends that you change the default password immediately after the initial login.
- If the server you installed is an additional server in a high availability setup, you can use the same credentials as the primary server to log in to the OpenEdge Command center dashboard.

The default email address for the super admin user is `admin@oecc.com`. You must change this default email address to a preferred one using the `reset superadmin` utility. Otherwise, you cannot add new user accounts. You can also use the `reset superadmin` utility to optionally update the other super admin user details, including the password, first name, and last name. For more information, see [Reset super admin user details](#) on page 103 and [RESETSUPERADMIN utility](#) on page 127.

Built-in database user account

The built-in user credentials created for internal use in the OpenEdge Command Center database are as follows:

- Username: dbadmin
- Password: `<encrypted password>`

You can update the password for this built-in OpenEdge Command Center database user account by using the `resetoeccdbpassword` utility. For details, see [RESETOECCDBPASSWORD utility](#) on page 126.

Log in to OpenEdge Command Center dashboard

The display of the OpenEdge Command Center login page depends on the active authentication mode:

- In local authentication mode, the login page displays the **Username** and **Password** fields.
- In IdP authentication mode, the login page displays the fixed **Sign in with OAuth** button instead of the local **Username** and **Password** fields.

Log in using local authentication

To log in to OpenEdge Command Center dashboard, perform the following steps:

1. Enter `https://host[:port]` in the browser address bar.

The host is the name of a system on which OpenEdge Command Center is installed, and the optional port number is the web server port. By default, this port is 8000. A sign-in form appears, prompting you to enter your login credentials.

Note: If OpenEdge Command Center operates in high availability mode, and you are using a load balancer, then omit the port number when you specify the load balancer host name.

2. If you are logging in for the first time, enter your login credentials and click **Sign in**.
3. After you log in to OpenEdge Command Center for the first time, you must establish initial configurations before you can use it.

Log in using IdP authentication

To log in to OpenEdge Command Center in IdP authentication mode, perform the following steps:

1. Enter the OpenEdge Command Center URL in the browser and click the **Sign in with OAuth** button.

The log in page of your identity provider appears.

2. If you already have an active single sign-on session with the identity provider, you are returned to OpenEdge Command Center without being prompted for credentials.
3. If you do not have an active single sign-on session, enter your credentials on the identity provider log in page and complete the multi-factor authentication that your organization enforces.

The identity provider returns you to the OpenEdge Command Center dashboard. Your permissions are determined by the app roles that your identity provider administrator assigns to you.

If your session expires, you are returned to the login page. When the single sign-on session of the identity provider is still valid, you sign in again without entering your credentials.

Note: In IdP authentication mode, password management, including forgotten password recovery and multi-factor authentication are handled by the identity provider.

On OpenEdge Command Center, the menu bar consists of the following functional options:

- Dashboard
- OECC Agents
- OpenEdge Databases
- PASOE Instances
- ABL Applications
- Users
- OECC System Settings
- Email Settings

Note: If OpenEdge Command Center detects no activity for more than 30 minutes, your session times out.

Upon first login to the OpenEdge Command Center console dashboard, Progress recommends changing the default admin password. For information about changing the default password after the first login, see [Change or reset passwords](#) on page 31.

Switch authentication modes

You use the SWITCHAUTHMODE utility to store authentication configurations on the OpenEdge Command Center server and to select the configuration that is active. Adding a configuration does not change the running authentication mode. Only activation changes the mode, and the change takes effect after you restart the OpenEdge Command Center server.

SWITCHAUTHMODE prompts you for the configuration details, one at a time. You enter the configuration name, provider, issuer URL, client ID, client secret, redirect URI, scope, and optional roles claim path, when prompted. The client secret is not echoed.

Prerequisites

Before you begin, ensure that the following conditions are met:

- OpenEdge Command Center is registered as an application in the identity provider, the required app roles are created and assigned, and the access token is in JWT format. For example, configure a custom API scope in Microsoft Entra ID or a custom authorization server in Okta. Ensure that a refresh token is enabled.
- Users that are assigned the `role:oecc/oecc_admin` app role can view all agents and their partition IDs, but do not automatically see full details for every agent. Users assigned only agent roles can view and manage only the agents for those roles. Users without any OpenEdge Command Center app role can sign in but cannot view OpenEdge Command Center or agent details.
- You can sign into the OpenEdge Command Center server host with root privileges on Linux, or with Administrator privileges on Windows.
- You can restart the OpenEdge Command Center server.

Switch from local authentication to IdP authentication

Before you begin, ensure that the following conditions are met:

- OpenEdge Command Center is registered as an application in the identity provider. The required app roles are created and assigned to users. The identity provider must include assigned OpenEdge Command Center app roles in the `access_token`, which must be in JWT format. Ensure that a refresh token is enabled. For provider-specific information, see the documentation for your identity provider.
- Users assigned the `role:oecc/oecc_admin` app role can view all agent names and their partition IDs. This role does not automatically grant full access to every agent. Users must also be assigned the appropriate agent roles to view and manage those agents. Users without an OpenEdge Command Center app role can sign in but cannot view OpenEdge Command Center or agent details.
- You can sign in to the OpenEdge Command Center server host with root privileges on Linux or Administrator privileges on Windows.
- You can restart the OpenEdge Command Center server.

To activate an identity provider configuration, perform the following steps:

1. On the OpenEdge Command Center server host, open a command-line interface with root privileges on Linux or Administrator privileges on Windows. Go to the `<OEC-Server-Installation-Directory>\Server\utils` directory.
2. To add the identity provider configuration, run the following command:

```
switchauthmode --add
```

The utility prompts you to enter the configuration name, provider, issuer URL, client ID, client secret, redirect URI, scope, and optional roles claim path. Enter the values supplied by your identity provider Administrator. The client secret is not displayed as you type it.

Adding a configuration does not change the active authentication mode. Users continue to sign in locally until you activate the configuration.

3. To confirm that the configuration is stored and that the local configuration is still active, run the following command:

```
switchauthmode --list
```

4. Activate the identity provider configuration. Replace <configuration-name> with the name you entered when adding the configuration:

```
switchauthmode --activate "<configuration-name>"
```

5. Go to the <OECC-server-installation-directory>\Server directory, and restart the OpenEdge Command Center server:

```
oeccserver stop oeccserver start
```

In a high-availability deployment, restart every server so that each server reads the active configuration at startup.

6. Open the OpenEdge Command Center URL. Confirm that the login page displays the **Sign in with OAuth** button and no longer displays the user name and password fields.
7. Users that are assigned the role:oecc/oecc_admin app role can view agent names and partition IDs in the **Agents** grid. The identity provider administrator can use a partition ID when creating the corresponding agent app roles.

Return to local authentication

You can roll back OpenEdge Command Center to local authentication if the external identity provider (IdP) is unavailable, misconfigured, or prevents users from signing in. This rollback applies to the entire OpenEdge Command Center deployment. In a multiple-node deployment, the rollback and restart must be completed on all servers. Individual users cannot switch authentication modes for themselves.

The IdP configuration is not deleted during the rollback. It remains stored and inactive, allowing you to reactivate the IdP authentication later, without entering the configuration values again.

Prerequisites

Before you begin, ensure that the following conditions are met:

- You can sign into the OpenEdge Command Center server host with root privileges on Linux, or with Administrator privileges on Windows.
- You can restart the OpenEdge Command Center server and all other servers in a multiple-node deployment.

To restore local authentication, perform the following steps:

1. On the OpenEdge Command Center server host, run SWITCHAUTHMODE --LIST to confirm the exact name of the local configuration.
2. To activate local authentication, run SWITCHAUTHMODE --ACTIVATE LOCAL on the OpenEdge Command Center server host.

This utility sets the local configuration as the active configuration and keeps the IdP configuration stored and inactive.

3. Restart the OpenEdge Command Center server:

```
oeccserver stop  
oeccserver start
```

In a multiple-node deployment, restart every node because each node reads the active configuration at startup.

4. Open the OpenEdge Command Center URL and sign in as a restored local user by entering user name and password.

The login page displays the **Username** and **Password** fields, instead of the **Sign in with OAuth** button of the IdP.

5. To reset the password, click **Forgot password**.
6. Run `SWITCHAUTHMODE --LIST` and confirm that the IdP configuration is still stored and is now inactive. You can activate it later without entering its values again.

Change or reset passwords

Note:

- In local authentication mode, you can change or reset the password in OpenEdge Command Center. Whereas in IdP authentication mode, passwords are managed entirely by the identity provider administrator, therefore, the change password and forgot password pages are not available.
 - When you switch from IdP authentication to local authentication, all restored local users, including the administrator, must set a new password at first sign-in, by using the **Forget password** link. For more information, see [Switch from external IdP authentication to local authentication](#). Alternatively, the administrator can use the `RESETFIRSTUSER` utility to reset a password.
-

In local authentication mode, you can change or reset your password after you have logged in to OpenEdge Command Center.

To change your password:

1. On the OpenEdge Command Center, go to your user account option on the top-right corner.
2. Select **Change password**.
3. On the **Change Password** page, enter information in the following fields:

- **Old Password**
- **New Password**
- **Confirm Password**

Note that the password you create must have:

- Between 8-40 characters
- A mixture of uppercase and lowercase letters
- At least one numeral
- At least one of the special characters: ! @ \$ % ^ & () _ + = [] |

In addition, a password must not contain the following:

- Your username
- A previously used password

4. Click **Change Password**.

Configure email settings

To create a user, you must first set up the email settings. Email settings can be set up by a super administrator or an administrator.

OpenEdge Command Center supports all standard mail servers (Gmail, Hotmail, Yahoo, and others).

Note: New users can be created only after the email settings are configured.

To configure email settings:

1. Go to **Email Settings**.
2. On the **Email Settings** page, enter information in the following fields:
 - **SMTP Host name**
 - **SMTP Port name**
3. Select the **Secure Connection** checkbox to enable a secure connection (HTTPS).
4. Select the **Allow Only Trusted Certificates** checkbox to enable using trusted certificates. This checks for valid certificate. It is enabled by default.
5. Alternatively, use the **STARTTLS Options** to customize your secure connection.
 - Ignore STARTTLS
 - Use STARTTLS when available

- Require STARTLS
6. Select the **SMTP Authentication** checkbox to enable SMTP authentication and enter the following:
 - **Mail server (SMTP) username**
 - **Mail server (SMTP) password**
 7. Enter a valid email address in the **Default email sender** field.

Create a new user

Note: In IdP authentication mode, you cannot create users, edit user details, or assign roles in OpenEdge Command Center because the identity provider (IdP) administrator creates and manages users in the IdP portal. The IdP administrator must assign the required app roles in the IdP portal, and OpenEdge Command Center uses these role assignments to control user access.

In local authentication mode, during installation, the first user to be set up is given the super administrator role

Note: New users can be created only after the email settings are configured. See here, [Configure email settings](#) on page 32.

To add a new user:

1. Go to **Users**, and click **New User**.
 2. Enter the following information:
 - **First name**
 - **Last name**
 - **Username**
 - **(Optional) Description**
 3. Choose a role from the drop-down list:
 - **Administrator**—An administrator can create users and perform all actions, but cannot assign a super administrator role.
 - **No access**—If a user is set to this role, then he or she cannot access the OpenEdge Command Center.
-

Note: Information about character length for each field is available inline. Ensure that the details you enter conform to the rules.

4. An email is sent to the email address entered, and you are prompted to set your login credentials.

After a new user is created, you can view the following information on the **Users** page on the dashboard:

- Username
- Role

- Email
- Description (whether the user was created during installation or added as a new user)
- Created at
- Last login

By default, columns are sorted by time stamps, but you can sort them as desired.

After you create a new user account, the system does not grant the user access to any remote resources by default. To enable access, you must create a corresponding user in the Authorization server and assign roles using the Authorization server REST APIs. For more information, see [Create users and manage roles using Authorization server](#) on page 73 and *Authorization Server REST API Reference* guide.

Note: Before using the Authorization server REST APIs, you must authenticate with the OpenEdge Command Center OpenAPI to obtain an authorization token. This token is required to access the Authorization server OpenAPI. For detailed steps, see [Log in to Authorization server](#) on page 74 .

Until you assign roles, the user can access only basic user management operations, such as viewing users and updating their own profile.

Edit user details

Note: In identity provider (IdP) authentication mode, you cannot create users, edit user details, or assign roles in OpenEdge Command Center because the IdP administrator creates and manages the users. The IdP administrator must manage user attributes and assign the required app roles in the IdP portal. OpenEdge Command Center uses these role assignments to control user access.

In local authentication mode, you can edit the first name, last name, description, email, and role of a user. To update the email address and role, you receive a verification code through email, for added security. The other updates are applied immediately.

In local authentication mode, to update the details of a user:

1. In the OpenEdge Command Center console, click **Users**. The **Users** page appears.
2. Select the user whose details you want to edit. The **Edit User** page appears with the following fields enabled for editing:
 - **First Name**
 - **Last Name**
 - **Email**
 - **Description**
 - **Roles**
3. Edit the required fields and click **Save**.
 - If the changes are for first name, last name, and description, the updates are applied immediately.
 - If the changes are for email address and role, the **Identity Verification** dialog box appears.

4. In the **Verification Code** field, enter the code that you received on your email to complete the verification.
5. Click **Verify**. After successful verification, the updates are applied.

Manage OpenEdge Command Center agents

The **OECC Agents** page lets you view and manage the agents linked to your OpenEdge deployments.

In IdP authentication mode, the **OECC Agents** page includes a **Partition ID** column, whose value in this column identifies the partition to which the agent belongs. An IdP administrator uses this value to create the matching app roles for the agent and grant users access to that agent.

This page displays the following details of the OpenEdge Command Center agents:

- **Name**—Name of the agent.
- **Labels**—Labels that you assign to the agent to categorize and filter it.
- **Host Name**—Host name of the system where the agent is running.
- **OS Name**—Name of the operating system on which agent is running.
- **OS Version**—Version of the operating system on which agent is running.
- **OS Family**—Classification of the operating system on which agent is running.
- **Version**—Version of the agent.
- **Status**—Indicates whether the agent is running or offline.
- **OECC Agent Start Time**—The date and time when the agent last started.
- **IP Address**—The IP address of the system where the agent runs.

From this page, you can perform the following tasks:

- Search and filter agents using defined parameters
- Add or edit labels to organize agents
- Edit agent names
- Unregister agents when necessary

Note:

- In local authentication mode, access is controlled by the roles assigned in OpenEdge Command Center.
 - In IdP authentication mode, access is controlled by the app roles assigned to the user in the identity provider. A user who has access to only a specific agent can perform tasks only for that agent.
-

Add agent label

You can add agent labels for the agents that are configured on OpenEdge Command Center. Adding labels allows you to easily filter development, test, or production components of the agents.

Add agent labels:

1. Go to **OECC Agents**.
2. Select an agent, and click **Actions**.
3. Select **Add Labels** from the drop-down list.
4. Enter the label in the **Labels** field, and click **Save**.

Note:

- You can enter up to five labels per agent.
 - The maximum number of characters allowed in a label is 40. The supported characters are alphanumeric, dot (.), underscore (_), and hyphen (-). Other special characters are not supported.
-

Update agent name and label

To update OpenEdge Command Center agent name and labels:

1. Go to **OECC Agents**.
2. Click the agent. The **Edit Agent** page is displayed.
3. On the **Edit Agent** page, update the required fields:
 - **Name**—Name of the agent host
 - **Labels**—Labels associated with corresponding agent
4. Click **Save** to update the agent information.

You can view agent labels on the **OECC Agents** page.

Search for and filter agents

You can search for and filter Command Center agents based on the defined parameters.

To search for Command Center agents:

1. Go to **OECC Agents**.
2. Place your cursor in the **Type here to filter** space on the search bar above the listed agents.
3. Select one of the following options from the drop-down list, and then choose one of the available criteria that you want to filter by:
 - **Agent Name**
 - **Labels**
 - **Host Name**
 - **OS**
 - **Version**
 - **Status**
 - **IP Address**
4. Enter the required information. For example, if you select **Labels**, then enter the label name that you want to filter by. OpenEdge Command Center supports the use of the asterisk (*) wildcard character in filter specifications.

The filtered agents are displayed.

Unregister an agent

You can unregister an agent to remove it from OpenEdge Command Center when the agent is offline.

To unregister an agent, perform the following steps:

1. In the **OpenEdge Command Center** console, click **OECC Agents**.

The **OECC Agents** page appears.

2. Select the check box next to the agent with an OFFLINE status that you want to unregister.
3. On the **Actions** menu, click **Unregister OECC agent**.

The **Unregister Agents** popup window appears.

4. Click **Unregister Agents**.

A notification confirms that the agent was successfully unregistered. Optionally, you can refresh the notification and view the outcome of the operation by clicking the **bell** icon.

Assign agent permissions in IdP authentication mode

The **OECC Agents** grid includes a **Partition ID** column. In IdP authentication mode, IdP administrator uses this partition ID value to authorize users for a specific agent.

When an agent connects to OpenEdge Command Center, it automatically creates the roles for that agent. However, you must create the corresponding app roles manually in the IdP portal.

1. In the **OECC Agents** grid, locate the agent and copy its **Partition ID** value.
2. In the IdP portal, create an app role for each level of access that you want to grant for the agent, and then replace `<partitionId>` with the copied value:

```
role:agent/<partitionId>/agent_admin  
role:agent/<partitionId>/agent_resource_user
```

3. Similarly, set the value of each app role, including case. Set the display name to a name that identifies the agent, such as the agent name. A role value that does not match an OpenEdge Command Center role is ignored and grants no permissions.
4. Users assigned the `role:authz/authz_admin` and `role:agent/<partitionId>/agent_admin` can create and manage custom roles.
5. Assign the users or groups that manage the agent to the new app roles.

At the next sign-in, or after the user refreshes the session, the identity provider must include the assigned agent role in the user's `access_token`. OpenEdge Command Center uses that role to authorize the user to work with the agents assigned to that partition.

Note: A user assigned the `role:oecc/oecc_admin` app role can view all agents and their **Partition ID** values in the **OECC Agents** grid.

IdP navigation and profile changes

When IdP authentication mode is active, the OpenEdge Command Center interface changes as follows:

- The **Users** grid is not available in the menu bar, because user accounts are managed in the identity provider and local user management is disabled.
- The header profile menu displays the signed-in user details from the IdP token, and does not include the **Change Password** option.
- The **Email** grid is not available in the menu bar.

Manage OpenEdge databases

This topic describes how you can manage the OpenEdge databases using OpenEdge Command Center.

The **OpenEdge Databases** tab enables you to manage OpenEdge databases. When you click this tab, the **OpenEdge Databases** page appears. On this page, you can view both the running databases that are automatically discovered and the database connections that you have previously added. You can add new database connections, start or stop databases, and edit or remove existing database connections. You can view the following details of the OpenEdge databases:

- **Name**—Name of the database.
- **Labels**—Labels that are assigned to the database.
- **Path**—Complete path of the database to which you are connected.
- **Agent**—Name of the OpenEdge Command Center agent.
- **Status**—Indicator of whether the database is running.
- **Port/Service Name**—Port number or service name of the database.
- **Version**—Version of the database.
- **Host Name**—Name of the machine where the database is hosted.
- **OpenEdge Installation**—The OpenEdge installation where the database is present.

The details of the databases are organized in columns. You can view or hide specific columns using the **Add/Remove Columns** button. When you click this button, the **Add/Remove Columns** picker appears, displaying a list of column names along with toggle buttons. You can use the toggle button next to each column name to view or hide the corresponding column on the page. You can refresh the content on the page by clicking **Refresh**.

The **OpenEdge Databases** page also lets you log in to any running OpenEdge database to view its schema, users, and user roles details.

Filter OpenEdge databases

You can quickly search and filter the databases using the **Type here to filter** field. It allows you to select one of the following options from the drop-down list and choose the available criteria to filter by:

- **Name**
- **Status**
- **Labels**
- **Host Name**
- **Version**
- **Agent Name**

Add a new OpenEdge database connection

This section describes how you can add an OpenEdge database connection using OpenEdge Command Center.

To add an OpenEdge database connection, perform the following steps:

1. In the OpenEdge Command Center console, click **OpenEdge Databases**.

The **OpenEdge Databases** page appears.

2. Click **+ Add New Connection**.

The **New Database Connection** page appears.

3. Specify the following OpenEdge database details:

In the following field . . .	. . . do the following
OpenEdge Installation	Select the OpenEdge installation where the database is present.
Database Folder Path	Specify the complete path of the database which you want to connect. For example, /usr/dlcwrk/mydb/database1.db.
Database Name	Displays the name of the database. It is automatically populated after you enter the complete path of the database.
Labels	Lets you distinguish between database resources. Select existing or create new labels to assign them to the database.
Port/Service Name	Specify the port number or service name for connecting to the database.
User ID	Specify a user ID with Database Administrator (DBA) privileges to connect to the database. Note: The User ID can contain only alphanumeric characters and certain special characters: underscore (_), dash (-), and number sign (#).
Password	Specify a password for the user ID.
Confirm Password	Re-enter the password.
Other Parameters	Specify the list of optional parameters, separated by space, to connect to the database. For example, to specify the Maximum Clients per Server and the Maximum Servers parameters, enter the following: -Ma 10 -Mn 1 Note: • If the OpenEdge database is running, the parameters used while starting the database are appended to the parameters you enter. • If the OpenEdge database is stopped, the -db and -S parameters are prepended by default to the parameters you enter.

4. Optional. Click **Test Connection** to verify whether the values entered in the respective fields are correct
5. Click **Save**.

A notification popup window appears, displaying a message that the database connection is in progress. Optionally, you can refresh the notification and view the outcome of the operation by clicking the bell icon. After the connection is successfully established, the **OpenEdge Databases** page displays the database details and its current status.

Start or stop OpenEdge databases

From the OpenEdge Command Center console, you can start or stop one or more OpenEdge databases, as follows:

1. In the OpenEdge Command Center console, click **OpenEdge Databases**.

The **OpenEdge Databases** page appears.

2. Select the check box next to each database that you want to start or stop.

Note: You can start only those databases whose status is displayed as **STOPPED**, and stop only those databases whose status is displayed as **RUNNING**.

3. To start or stop one or more databases, perform either of the following set of steps respectively:

- To start one or more databases, from the **Actions** menu, click **Start**.

A notification popup window appears, displaying a message that the start operation is in progress. Optionally, you can refresh the notification and view the outcome of the operation by clicking the bell icon. After the connection is established successfully, the database details appear on the **OpenEdge Databases** page, and the database status is displayed as **RUNNING**.

- To stop one or more databases, from the **Actions** menu, click **Stop**. In the popup window, type **Stop** and click **Stop**.

A notification popup window appears, displaying a message that the start operation is in progress. Optionally, you can refresh the notification and view the outcome of the operation by clicking the bell icon. After the connection is established successfully, the database details appear on the **OpenEdge Databases** page, and the database status is displayed as **STOPPED**.

Edit OpenEdge database connections

From the OpenEdge Command Center console, you can edit OpenEdge database connections, as follows:

1. In the OpenEdge Command Center console, click **OpenEdge Databases**.

The **OpenEdge Databases** page appears.

2. Click the name of the OpenEdge database whose connection details you want to update.

The **Edit Database Connection** page appears.

3. Update the values in any or all of the following fields, as required.

Field	Description
Labels	Lets you distinguish between database resources. Select existing or create new labels to assign them to the database.
Port/Service Name	Specify the port number or service name for connecting to the database.
User ID	Specify a user ID with Database Administrator (DBA) privileges to connect to the database. Note: The User ID can contain only alphanumeric characters and certain special characters such as underscore (_), dash (-), and number sign (#).
Password	Specify a password for the user ID.
Confirm Password	Re-enter the password.
Other Parameters	Specify the list of optional parameters, separated by space, to connect to the database. For example, to specify the Maximum Clients per Server and the Maximum Servers parameters, enter the following: -Ma 10 -Mn 1

- Optional. Click **Test Connection** to verify your entered values are correct.

Note: **Test Connection** is available only if the database is in running status.

- Click **Save**.

A notification popup window appears, displaying a message that the database connection details are updated successfully. Optionally, you can refresh the notification and view the outcome of the operation by clicking the bell icon.

Remove OpenEdge database connections

From the OpenEdge Command Center console, you can remove one or more OpenEdge database connections, as follows:

- In the OpenEdge Command Center console, click **OpenEdge Databases**.
The **OpenEdge Databases** page appears.
- Select the check box next to each database connection that you want to remove.
- On the **Actions** menu, click **Delete Connection**. The **Delete Connections** popup window appears.

4. In the **Type delete to confirm** field, type **delete**.
5. Click **Delete Connections**.

A notification popup window appears, displaying a message that the selected database connections are removed. Optionally, you can refresh the notification and view the outcome of the operation by clicking the **bell** icon. After the database connections are removed successfully, the corresponding database details no longer appear on the **OpenEdge Databases** page.

View schema, users, and roles of a selected database

From the **OpenEdge Command Center** console, you can view details of schema, users, and roles of a running database, as follows:

1. In the OpenEdge Command Center console, click **OpenEdge Databases**.
The **OpenEdge Databases** page appears.
2. Select the checkbox next to the running OpenEdge database for which you want to view the details. The **<database_name>** pane appears at the bottom of the page, displaying three disabled tabs: **Schema**, **Users**, and **Roles**.
3. Log into the database by clicking **Login**. The **Database Authentication** dialog box appears.
 - a) In the **Database User ID** field, type the username.
 - b) In the **Database Password** field, type the password.
 - c) Click **Authenticate**.
The **Schema**, **Users**, and **Roles** tabs are enabled.
4. Click each of the **Schema**, **Users**, or **Roles** tab to view the corresponding details of the database.
 1. To view the details about database schema and their tables and columns, click the **Schema** tab. The following sections appear:
 - **Tables**: Lists the tables within the database.
 - **Columns**: Lists the columns that are associated with the table.
 - **Details**: Lists the column details, such as field, datatype, display format, and default value.You can quickly search for tables and columns by clicking the search icon next to the corresponding section title. You can get the latest data from the OpenEdge database by clicking **Refresh**.
 2. To view the users who have access to the database, and their usernames and associated user-defined and system-defined roles, click the **Users** tab. The following sections appear:
 - **Users**: Lists all the users within the database.
 - **Details**: Lists the user details, such as username and roles assigned to the user.You can quickly search for users by clicking the search icon next to the **Users** section title. You can get the latest data from the OpenEdge database by clicking **Refresh**.
 3. To view details about various user-defined and system-defined roles, click the **Roles** tab. The following sections appear:
 - **Roles**: Lists the user-defined and system-defined roles.

- **Details:** Lists the role details, such as created by, type, and description.

You can quickly search for roles by clicking the search icon next to the **Roles** section title. You can get the latest data from the OpenEdge database by clicking **Refresh**.

Configure and manage PAS for OpenEdge instances

After you have configured an agent, the PAS for OpenEdge instances associated with that agent become discoverable on OpenEdge Command Center. The instances are available on the **PAS for OpenEdge Instances** page on the OpenEdge Command Center dashboard.

You can view the following information about the PAS for OpenEdge instances:

- **Name**—Name of the PAS for OpenEdge instance.
- **Labels**—All labels that are assigned to that server; for example: test, production, or development.
- **Status**—Indicator of whether the PAS for OpenEdge instance is running.
- **Host Name**—Name of the machine where the instance is hosted.
- **Install Path**—Location of where the instance is installed on the agent.
- **Version**—Version of the PAS for OpenEdge instance.
- **Catalina Base**—Path of the *CATALINA_BASE* directory.
- **IP Address**—IP address of the PAS for OpenEdge instance host.
- **Agent**—Name of the PAS for OpenEdge instance agent

Configure PAS for OpenEdge instances

This topic describes how you can configure PAS for OpenEdge instances using the OpenEdge Command Center console.

Filter PAS for OpenEdge instance properties

The **Edit PAS For OpenEdge Instance** page displays many configuration properties. You can use a filter to easily locate the property you want to modify.

To filter the properties of the PAS for OpenEdge instance:

1. In the OpenEdge Command Center console, click **PASOE Instances**.

The console displays a page that lists all PAS for OpenEdge instances that are associated with all currently configured OpenEdge Command Center agents.

2. Click the name of the PAS for OpenEdge instance whose configuration you want to modify.

The **Edit PAS For OpenEdge Instance** page is displayed.

3. In the search field, enter the name, label, or section header of the property that you want to locate. You can also enter the property name as specified in the *catalina.properties* file. For example, enter `psc.as.executor.maxthreads` (as it appears in the *catalina.properties* file) to locate the **Max Threads** property. Only the **Max Threads** property field is displayed on the page.

Edit PAS for OpenEdge instance properties

This topic explains how to edit the properties of a PAS for OpenEdge instance on the **Edit PAS for OpenEdge Instance** page:

To edit the properties of a PAS for OpenEdge instance:

1. In the OpenEdge Command Center console, click the **PASOE instances** tab.

The console displays a page that lists all PAS for OpenEdge instances that are associated with all currently configured OpenEdge Command Center agents.

2. Click the name of the PAS for OpenEdge instance whose configuration you want to modify.

The **Edit PAS for OpenEdge Instance** page is displayed.

3. In the **Info** section, update values for the following properties:

Field	Description
Instance Name	Name of the PAS for OpenEdge instance. Note: The PAS for OpenEdge instance name is case-sensitive and must be at least 5 characters long. It can include any character except periods (.) or square brackets ([]). The name must be unique among all configured PAS for OpenEdge instance names.
Labels	One or more labels assigned to the instance. Use the drop-down box to select one or more labels, or create one or more new labels to assign.

4. Click **HTTPS** to expand the HTTPS connection properties section.

Provide values for the following properties:

Property	Default value	Description
HTTPS Connector	ON	Use the ON/OFF toggle switch to enable or disable the HTTPS connector respectively.
Port	8443	Specifies the HTTPS protocol connector port number.
Connection Timeout	20000	Specifies the time, in milliseconds, to wait between the establishment of a TCP (TLS) connection by a client and the arrival of the first HTTPS request.
SSL Session Timeout	86400	Specifies the time, in seconds, after which the TLS session ends.
Server Key Alias	test	Specifies the alias name of the keystore entry holding the server's private key and public key certificate.

Property	Default value	Description
Server Key Password	password	Specifies the password to use when accessing the TLS keystore.
Store Type	PKCS12	Specifies the type of Java keystore format used by Apache Tomcat keystore.
Max Queue Size	100	Specifies the maximum size of the HTTPS connector message queue.
Max Connections	-1	Specifies the maximum number of TCP socket connections per HTTPS connector.
Client Authentication	none	Specifies whether to enable or disable TLS client authentication by the HTTPS connector. The options are required , none , and optional . The default is none.
Certificate Trust Type	JKS	Specifies the type of Java certificate store format used by <code>tomcat-certstore.jks</code> that holds the root or intermediate CA certificates. These certificates are used to validate the clients using TLS client authentication.
Certificate Store Password	password	Specifies the Java certificate store password used to access <code>tomcat-certstore.jks</code> that holds the root or intermediate CA certificates. These certificates are used to validate the clients using TLS client authentication.
Encryption Protocol	TLSv1.2	Specifies the HTTPS protocol selected for secure communication. If required, you can also edit this value in the <code>catalina.properties</code> file.
Enabled Ciphers	ALL	Specifies the list of cipher suites enabled for secure communication. This property can either be all the cipher suites or a comma-separated list of cipher suites supported by JSSE.
Bind On Init	false	Check box to control when to bind the socket used by the connector.
Compression	on	Check box to enable GZIP compression for HTTPS transports.

- Click **HTTP** to expand the HTTP connection properties section.

Provide values for the following properties:

Property	Default value	Description
HTTP Connector	ON	Use the ON/OFF toggle switch to enable or disable the HTTP connector respectively.
Port	8090	Specifies the HTTP protocol connector port number.
Connection Timeout	20000	Specifies the time, in milliseconds, to wait between the establishment of a TCP connection by a client and the arrival of the first HTTP request.
Max Connections	-1	Specifies the maximum number of TCP socket connections per HTTP connector.
Max Queue Size	100	Specifies the maximum size of HTTP connector message queue.
URI Encoding	ISO-8859-1	Specifies the character encoding used to decode the URI bytes after decoding the URL. Note: This value is set to the default value of the Apache Tomcat server, and it affects both the HTTP and HTTPS connectors.
Bind On Init	false	Check box to control when to bind the socket used by the connector.
Compression	on	Check box to enable GZIP compression for HTTP transports.

6. Click **Server Options** to expand the Server Options section.

Provide values for the following properties:

Property	Default value	Description
Shutdown Port	—	Specifies the port number to shutdown the server that is running. The values can range from 1024 to 65535. To disable the shutdown port, set the value to -1. You must specify a value in Windows, but this property is optional in UNIX.

Property	Default value	Description
Shutdown Password	SHUTDOWN	Specifies a private shutdown port access code to prevent one server instance from being shutdown by anyone. Note: If the shutdown port is specified, you must change this value to avoid insecure configuration.
Stuck Session Valve	ON	Specifies whether the stuck session valve is enabled to monitor and manage sessions that become unresponsive or exceed a defined execution time.
Stuck Thread Threshold	600	Specifies the maximum amount of time, in seconds, an active HTTP request can be running before it is considered stuck and is reported in the server log file.
Max Threads	300	Specifies the maximum number of OS process threads the PAS for OpenEdge Server can use.
Minimum Spare Threads	10	Specifies the number of spare threads the server reserves for future client operations.
Web Application Directory	webapps	Specifies the location of the directory where the web applications are to be deployed. If not specified, then the default directory of the Apache Tomcat server, webapps is used. If a relative path is specified, it must be relative to the instance's root directory. If an absolute path is used, it must conform to a single platform type.
Common Library Path	—	Specifies a comma-delimited list of library paths. For example, local/server/common/*.jar, local/server/common/x.jar. The delimiter is platform dependent, and the format of the path must conform to the common library path format of the Apache Tomcat server (https://tomcat.apache.org/tomcat-9.0-doc/class-loader-howto.html).

Property	Default value	Description
Auto Deploy WAR Files	false	Select the checkbox to enable the security settings used by the Apache Tomcat server for deploying web applications. The default value depends on whether the PAS instance is created as a development or a production security configuration. In a production configuration, the default is false. In a development configuration, the default is true.
Unpack WAR Files	true	Select the checkbox to allow unpacking of web archive (.war) files when the PAS for OpenEdge instance is started.
Version Logger	ON	Use the ON/OFF toggle to enable or disable logging the command line arguments passed to Java when the Apache Tomcat server is started.
Log Command Line Arguments	true	Select the checkbox to log the command line arguments passed to Java when the Apache Tomcat server is started.
Log Environment Variables	false	Select the checkbox to log the current environment variables when the Apache Tomcat server is started.
Log Java System Properties	false	Select the checkbox to log the current Java system properties when the Apache Tomcat server is started.

7. Click **Request Options** to expand the Request Options section.

Provide values for the following properties:

Property	Default value	Description
Max POST Request Size	2097152	Specifies the maximum size, in bytes, of a connector's POST message body.
Max Pipeline Requests	100	Specifies the maximum number of pipelined HTTP keepalive requests before the TCP socket to the client is closed.

Property	Default value	Description
Message Timeout	10000	Specifies the maximum time, in milliseconds, to wait for asynchronous messages to complete.
Compression Minimum Size	2048	Specifies the minimum size, in bytes, of message that will be compressed, in bytes. This property is applicable to both HTTPS and HTTP transports.
Compression MIME Types	text/html, text/xml, text/javascript, text/css, application/json	Specifies a comma-separated list of MIME types that must be compressed. This property is applicable to both HTTPS and HTTP transports.

8. Click **JVM Settings** to expand the JVM Settings section.

Set the JVM arguments to configure Java environment settings for the PAS for OpenEdge instance. For example, set `-XX:NewSize` and `-XXMaxNewSize` to specify the minimum and maximum heap size used in garbage collection.

9. Click **Logging Configuration** to expand the Logging Configuration section.

Provide values for the following properties:

Property	Recommended setting	Description
Catalina Logging Level	INFO	Sets log level for the core Apache Tomcat server. By default, the logging level is set to INFO, but you can change it to WARN, FINE, FINER, ERROR, DEBUG, or TRACE, as required.
OpenEdge ABL Logging Level	WARN	Sets the log level for ABL application (oeabl.war) Session Manager and Spring Security logging.
Authentication Logging Level	ERROR	Sets the log level for ABL application login event logging.

Property	Recommended setting	Description
Authorization Logging Level	ERROR	Sets the log level for ABL application URL access event logging.
OpenEdge STS Logging Level	WARN	Sets the log level for the OpenEdge Authentication Gateway server Security Token Service web application.

10. Click **MS Agent Log Configuration** to expand the Agent Log Configuration section.

Provide values for the following properties:

Property	Description
MS Agent Log Entry Types	Specifies the types of log entries to write to the log file specified by the Client Logging and DataServer Logging startup parameters. If <code>allowRuntimeUpdate</code> is set to <code>true</code> , then changes to the property are applied without restarting the PAS for OpenEdge instance
MS Agent Log Level	Specifies the logging level for each entry type. You can select values from 0 to 4. If <code>allowRuntimeUpdate</code> is set to <code>true</code> , then changes to the property are applied without restarting the PAS for OpenEdge instance.

For more information about the Agent Log Configuration properties, see "Troubleshoot problems with an instance" in *Manage Progress Application Server (PAS) for OpenEdge*.

11. After modifying properties, you must restart the PAS for OpenEdge instance for most of the changes to take effect. However, if `allowRuntimeUpdate` is set to `true`, then changes to the **MS Agent Log Entry Types** and **MS Agent Log Level** properties are applied without restarting the PAS for OpenEdge instance.

Clone a PAS for OpenEdge instance

You can clone a PAS for OpenEdge instance to quickly create identical PAS for OpenEdge instances and help you set up load balancing. You can create up to eight identical PAS for OpenEdge instances in a single clone operation. For cloning, the platform of the destination OpenEdge installations must be the same as the source PAS for OpenEdge instance. Also, if the source PAS for OpenEdge instance refers to external files or folders, the cloned instances may not work as expected.

To clone a PAS for OpenEdge instance:

1. In the OpenEdge Command Center console, click the **PASOE instances** tab.

The **PAS for OpenEdge Instances** page displays a list of all PAS for OpenEdge instances that are associated with all the currently configured OpenEdge Command Center agents.

2. Select the check box for the PAS for OpenEdge instance that you want to clone.
3. On the **Actions** menu, select **Clone**.

The **Clone PAS for OpenEdge Instance** page appears.

4. On the **Clone PAS for OpenEdge Instance** page, enter the following details:

In the following field . . .	. . . do the following
Instance Name	Specify a name for all the destination PAS for OpenEdge instances. By default, this value is the name of the cloned PAS for OpenEdge instance. However, you can specify a different name based on the requirements. This field is mandatory. Note: The PAS for OpenEdge instance name is case-sensitive. It must be unique among all configured PAS for OpenEdge instances and can include any character except periods (.) or square brackets ([]).
Instance Size	View the size of the cloned PAS for OpenEdge instance. This value is automatically calculated by OpenEdge Command Center.
Target Directory	Specify the directory on the destination OpenEdge installation where OpenEdge Command Center creates the cloned PAS for OpenEdge instance.
Labels	Use the drop-down list to select one or more labels, or create one or more new labels to assign. You can have a maximum of seven labels for a PAS for OpenEdge instance. All the labels of the source PAS for OpenEdge instance are automatically added to the field. If the total number of labels is less than seven, then a new label, <code>cloned_from_<source></code> is added to the field.
OpenEdge Installation	Use the drop-down box to select the destination OpenEdge installations for cloning the PAS for OpenEdge instance. You can select up to eight different OpenEdge installations as destinations for cloning. If you select a destination that already has a PAS for OpenEdge instance with the same name, the destination appears in red color, which indicates that the cloning operation cannot proceed. Selecting the Overwrite duplicate PAS for OpenEdge instance option removes the red color and allows you to proceed with the cloning.
Overwrite duplicate PAS for OpenEdge instance	Select the check box to overwrite an existing PAS for OpenEdge instance in the destination, which has the same name as the source PAS for OpenEdge instance.
Start PAS for OpenEdge instance after it is cloned	Select the check box to start the PAS for OpenEdge instances immediately after they are created in the destination installations.

5. Click **Clone Instance**.

The notifications in the OpenEdge Command Center console inform you about the progress of the cloning process. As the cloned PAS for OpenEdge instances are created, they are displayed on the **PAS for OpenEdge Instances** page. If you selected the **Start PAS for OpenEdge instance after it is cloned** check box, the newly cloned instances are started and their status changes to **RUNNING**.

If a cloned PAS for OpenEdge instance fails to start because of a port conflict in the destination machine, update the configuration of the cloned instance to use a unique port, from the **Edit PAS for OpenEdge Instance** page.

Manage ABL applications or ABL web applications

A PAS for OpenEdge instance can have several deployed OpenEdge ABL applications on it. In the OpenEdge Command Center, you can access, deploy, and undeploy your ABL and web applications on the PAS for OpenEdge instances tab.

Access your ABL applications or ABL web applications

In the OpenEdge Command Center, an ABL application is hosted on a PAS for OpenEdge instance and is accessible from a PAS for OpenEdge client. OpenEdge supports REST, WEB, SOAP, and APSV transport services for accessing the ABL application logic.

To access your ABL and web applications:

1. On the OpenEdge Command Center user interface, click the **PASOE Instances** tab on the side menu.
The PAS for OpenEdge instances page is displayed and shows a list of your PAS for OpenEdge instances that are associated with all currently configured OpenEdge Command Center agents.
2. Select the check-box next to the instance that hosts the ABL application or web application.
The **Applications** window is displayed on the lower part of the screen.
3. Select the ABL application that you want to view.
The web application name, its deployment path, number of services, and the status the four transports is displayed.
4. Click the ABL application to view or edit its properties. The **Edit ABL Application** page appears, displaying the properties of the corresponding ABL application. For more information, see [Edit ABL application](#) on page 68.

Note: The OpenEdge Command Center release does not support viewing independent web applications such as manager, oemanager, oedbg. However, the OpenEdge Command Center needs a manager application to deploy or undeploy applications. When you create a new PAS for OpenEdge instance, OpenEdge Command Center provides an option to deploy a manager application and define login credentials. It is highly recommended that you do not use the default Apache Tomcat credentials to enable security for your manager application.

Deploy ABL application

To deploy an ABL application on a PAS for OpenEdge instance:

1. In the OpenEdge Command Center console, click the **PASOE instances** tab.
The **PAS for OpenEdge Instances** page appears, listing all the instances associated with the currently configured OpenEdge Command Center agents.
2. Select the checkbox next to the instance on which you want to deploy the ABL application. The **Applications** pane appears at the bottom of the page.
3. Click **Deploy**. The **Deploy Application** dialog box appears.
4. Select the **ABL Application** option as the application type.
5. In the **Select Application File** field, click **Select File** to browse to the location of the OpenEdge Archive (OEAR) file containing the ABL application and select the file.

Note:

- You can deploy ABL applications as an OEAR file only. For information about the OEAR file, see "Create an OpenEdge Application Archive using tcman export" and "Create an OpenEdge Application Archive using an Ant Build" in *Manage Progress Application Server (PAS) for OpenEdge*.
- If you are deploying an OEAR file on an instance that already contains ABL applications or web applications with the same names as those included in the OEAR file, you must undeploy the existing applications first.
- The OEAR file does not need to have the same name as the ABL application.
- You can upload an OEAR file only if its size is less than or equal to 523, 239, 424 bytes (499 MB). If you want to upload a larger file, update the `system-config.json` file by adding the `maxFileUploadSize` parameter with appropriate value and restart the OpenEdge Command Center server. The maximum value you can set is 3, 221, 225, 472 bytes (3 GB). For more information, see "OpenEdge Command Center server configuration" in *Install the OpenEdge Command Center*.

6. Based on the status of the PAS for OpenEdge instance, the dialog box displays either the **Deploy** or **Deploy & Restart** button. The following table summarizes different scenarios:

If	Then	
	Deploy Application dialog box displays	Required action
Running	The Deploy & Restart button	Click Deploy & Restart .
Stopped	The Deploy button	Click Deploy .

Note: The deployment of the ABL application on the PAS for OpenEdge instance is independent of the deployment status of the Apache Tomcat Manager application.

A popup appears, displaying the notification that the deploy operation is successfully submitted. You can click the bell icon to refresh and view the deployment status. After the successful deployment, the ABL application appears in the **ABL Applications** section of the **Applications** pane.

Note: If an ABL application is deployed on a stopped PAS for OpenEdge instance, you must start the instance to access the ABL application services. For information on how to start a PAS for OpenEdge instance, see [Start or stop a PAS for OpenEdge instance](#) on page 60.

Deploy web application

To deploy a web application into an ABL application on a PAS for OpenEdge instance:

1. In the OpenEdge Command Center console, click the **PASOE instances** tab.

The **PAS for OpenEdge Instances** page appears, listing all the instances associated with the currently configured OpenEdge Command Center agents.

2. Select the checkbox next to the instance on which you want to deploy the web application. The **Applications** pane appears at the bottom of the page.

3. Click **Deploy**. The **Deploy Application** dialog box appears.
4. Select the **Web Application** option as the application type.
5. In the **Select ABL Application** field, type a new ABL application name or select an existing one from the list.

Note: If you type a new ABL application name, an ABL application with that name will be created and the web application will be deployed into it.

6. In the **Select Application File** field, click **Select File** to browse to the location of the WAR file containing the web application and select the file.

Note:

- If you are deploying a WAR file into an ABL application that already contains web applications with the same names as those included in the WAR file, you must undeploy the existing applications first.
- You can upload a WAR file only if its size is less than or equal to 523,239,424 bytes (499 MB). If you want to upload a larger file, update the `system-config.json` file by adding the `maxFileUploadSize` parameter with appropriate value and restart the OpenEdge Command Center server. The maximum value you can set is 3,221,225,472 bytes (3 GB).

For more information, see "OpenEdge Command Center server configuration" in *For more information, see "OpenEdge Command Center server configuration" in **Install the OpenEdge Command Center**.*

7. In the **Web Application Name** field, type the name of the web application.

The web application name is automatically detected from the WAR file and is displayed in the **Web Application Name** field, by default. You can change the application name, if needed.

8. Based on the status of the PAS for OpenEdge instance and deployment of the Apache Tomcat Manager application, the dialog box displays either the **Deploy** or **Deploy & Restart** button. The following table summarizes different scenarios:

If		Then	
Status of PAS for OpenEdge instance	Status of Apache Tomcat Manager Application	Deploy Application dialog box displays	Required action
Running	Deployed	The Enter Apache Tomcat Manager credentials section and the Deploy button	Enter the credentials and click Deploy .
Running	Not deployed	The Deploy & Restart button	Click Deploy & Restart .
Stopped	NA	The Deploy button	Click Deploy .

Note: If you enter incorrect credentials, the application deployment becomes a normal deployment, requiring the PAS for OpenEdge instance to restart.

A popup window appears, displaying the notification message that the deploy operation is successfully submitted. You can click the bell icon to refresh and view the deployment status. After the successful deployment, the web application appears in the **Web Application** section.

Undeploy ABL application

To undeploy an ABL application from a PAS for OpenEdge instance:

1. In the OpenEdge Command Center console, click the **PASOE instances** tab.
The **PAS for OpenEdge instance** page appears, listing all the instances associated with the currently configured OpenEdge Command Center agents.
2. Select the checkbox next to the instance from which you want to undeploy the ABL application. The **Applications** pane appears at the bottom of the page.
3. In the **ABL Applications** section, select the checkbox next to the ABL application that you want to undeploy.
4. From the **Actions** menu, select **Undeploy ABL Application**. The **Undeploy ABL Application** dialog box appears, displaying either the **Undeploy** or **Undeploy & Restart** button, based on the status of the PAS for OpenEdge instance and deployment of the Apache Tomcat Manager application.
5. Based on the status of the PAS for OpenEdge instance and deployment of the Apache Tomcat Manager application, the dialog box displays either the **Undeploy** or **Undeploy & Restart** button. The following table summarizes different scenarios:

If		Then	
Status of PAS for OpenEdge instance	Status of Apache Tomcat Manager Application	Undeploy ABL Application dialog box displays	Required action
Running	Deployed	The Enter Apache Tomcat Manager credentials section and the Undeploy button	Enter the credentials and click Undeploy .
Running	Not deployed	The Undeploy & Restart button	Click Undeploy & Restart .
Stopped	NA	The Undeploy button	Click Undeploy .

A popup window appears, displaying the message that the undeploy operation is successfully submitted. You can click the bell icon to refresh and view the undeployment status. After the successful undeployment, the ABL application no longer appears in the **ABL Applications** section of the **Applications** pane.

Undeploy web applications

To undeploy one or more web applications from an ABL application on a PAS for OpenEdge instance:

1. In the OpenEdge Command Center console, click the **PASOE instances** tab.
The **PAS for OpenEdge Instances** page appears, listing all the instances associated with the currently configured OpenEdge Command Center agents.
2. Select the checkbox next to the instance that contains the ABL application from which you want to undeploy the web applications. The **Applications** pane appears at the bottom of the page.
3. In the **ABL Applications** section, select the checkbox next to the ABL application that contains the web applications that you want to undeploy. The web applications associated with the ABL application appear in the **Web Application** section.
4. Select the checkbox next to each web application that you want to undeploy.
5. From the **Actions** menu, select **Undeploy Web Applications**. The **Undeploy Web Applications** dialog box appears.
6. Based on the status of the PAS for OpenEdge instance and deployment of the Apache Tomcat Manager application, the dialog box displays either the **Undeploy** or **Undeploy & Restart** button:

If		Then	
Status of PAS for OpenEdge instance	Status of Apache Tomcat Manager Application	Undeploy Web Applications Dialog Box displays	Required action
Running	Deployed	The Enter Apache Tomcat Manager credentials section and the Undeploy button	Enter the credentials and click Undeploy .
Running	Not deployed	The Undeploy & Restart button	Click Undeploy & Restart .
Stopped	NA	The Undeploy button	Click Undeploy .

A popup window appears, displaying the notification message that the undeploy operation is successfully submitted. You can click the bell icon to refresh and view the undeployment status. After the successful undeployment, the web application no longer appears in the **Web Applications** section.

Note: If you undeploy all the web applications from an ABL application, the ABL application is automatically undeployed from the PAS for OpenEdge instance.

Manage PAS for OpenEdge instances

This topic describes how you can manage PAS for OpenEdge instances using OpenEdge Command Center.

Create a PAS for OpenEdge instance

To create a PAS for OpenEdge instance:

1. In the OpenEdge Command Center console, click the **PASOE instances** tab.

The console displays a screen that lists all PAS for OpenEdge instances that are associated with all currently configured OpenEdge Command Center agents.

2. Click **New**.

The **New PAS Instance** screen is displayed. Use this screen to specify the following details:

In the following field . . .	. . . do the following
Instance Name	Specify a name for the PAS for OpenEdge instance. This is a mandatory field. Note: The PAS for OpenEdge instance name is case-sensitive. It can include any character except periods (.) or square brackets ([]). The name must be unique among all configured PAS for OpenEdge instance names.
OpenEdge Installation	Use the drop-down box to select the OpenEdge installation that corresponds to the PAS for OpenEdge instance that you want to create.
Labels	Use the drop-down box to select one or more labels, or create one or more new labels to assign.
Security Model	Specify one of the following security models: • Production If you select this model, you can make configuration changes to adjust security settings to ensure your application operates correctly in a production instance. • Developer Developer mode minimizes or eliminates any accessibility restrictions, so a developer can quickly use the product “out of the box” to develop, test, and debug applications.
Instance Directory	Specify the path of the PAS for OpenEdge instance. Note: If you do not provide the path of the instance directory, the PAS for OpenEdge instance is created in the working directory set during OpenEdge installation. The name of the instance is used as the name of the directory where the PAS for OpenEdge installation is created. The directory must not already exist (either when provided, or when using the default location).

In the following field . . .	. . . do the following
HTTP Port	Specify an unused port number to be associated with the PAS for OpenEdge instance HTTP port. By default, this is set to 8080. Note: Each new PAS for OpenEdge instance that you create uses the default configuration. However, the port number must be unique for each PAS for OpenEdge instance for the instance to operate properly. If you specify a port number that is used by another PAS for OpenEdge instance, you are prompted to confirm whether you want to use the port number.
HTTPS Port	Specify an unused port number associated with the PAS for OpenEdge instance HTTPS port. By default, this is set to 8443. Note: If you specify a port number that is used by another PAS for OpenEdge instance, you are prompted to confirm whether you want to use the port number.
Shutdown Port	Specify an unused port number for shutdown. If you are creating a PAS for OpenEdge instance on a Windows machine, this is a mandatory field. Note: If you specify a port number that is used by another PAS for OpenEdge instance, you are prompted to confirm whether you want to use the port number.
Login	Enter the login ID of the Apache Tomcat Manager web application that hosts the PAS for OpenEdge instance. If you are using the Apache Tomcat Web server shipped with OpenEdge, then the default login ID is tomcat.
Password	Enter the password of the Apache Tomcat Manager web application that hosts the PAS for OpenEdge instance. If you are using the Apache Tomcat Web server shipped with OpenEdge, then the default password is tomcat.

- If you want to start the PAS for OpenEdge instance immediately after creating it, make sure that the **Start PAS instance after it is created** is enabled.
- To enable online deployment of ABL applications for your PAS for OpenEdge instance, select the '**Deploy Manager application to enable online deployment**' option. If you select the option for a developer instance

of PAS for OpenEdge, all web applications, such as manager and oemanager, are deployed. If you select the option for a production instance of PAS for OpenEdge, only the Manager web application is deployed. OpenEdge Command Center uses the Manager application to load the application deployment status in PAS for OpenEdge.

5. Click **Create PAS Instance**.

Start or stop a PAS for OpenEdge instance

From the OpenEdge Command Center console, you can start or stop one or more local PAS for OpenEdge instances.

To start or stop one or more PAS for OpenEdge instances:

1. In the OpenEdge Command Center console, click the **PASOE instances** tab.

The console displays a screen that lists all PAS for OpenEdge instances that are associated with all currently configured OpenEdge Command Center agents.

2. Select the check box for each PAS for OpenEdge instance that you want to start or stop.

Note: You can start or stop a PAS for OpenEdge instance depending on what its current status is.

3. From the **Actions** menu, select **Start** or **Stop**, as appropriate.

A notification popup is displayed that shows the start or stop operation in progress.

4. Click the **Refresh** button to view the change in status of the corresponding PAS for OpenEdge instance.

Note: After you initiate a stop operation for multiple PAS for OpenEdge instances, a notification message is displayed indicating the stop operation is successfully initiated. You can see the success or failure of each PAS for OpenEdge instance stop operation from the separate notifications section.

Delete a PAS for OpenEdge instance

To delete one or more PAS for OpenEdge instances:

1. In the OpenEdge Command Center console, click **PASOE Instances**.

The console displays a screen that lists all PAS for OpenEdge instances that are associated with all currently configured OpenEdge Command Center agents.

2. Select the check box for each PAS for OpenEdge instance that you want to delete.

3. In the **Actions** menu, click **Stop**, if the instances are running.

4. In the **Actions** menu, click **Delete**.

A confirmation dialog box is displayed, prompting you to confirm your selection.

5. In the confirmation dialog box, beneath the field labeled **Type delete to confirm**, enter delete and click **Delete**.

Note: After you initiate a delete operation for a PAS for OpenEdge instance, a notification message is displayed indicating that the delete operation is successfully initiated. You can see the success or failure of each PAS for OpenEdge instance delete operation from the separate notifications section.

Obtain process details

On the **PAS for OpenEdge Instances** screen, you can view the following information about the Progress Application Server (PAS for OpenEdge instances):

- Name—Name of the PAS for OpenEdge instance.
- Labels—Any labels that are assigned to that server, for example: test, production, development.
- Status—If the PAS for OpenEdge instance is running.
- Hostname—Name of the machine where the instance is hosted.
- Install path—The location where the instance is installed on the agent.
- Version—The version of the PAS for OpenEdge instance.

To view the process details for a running PAS for OpenEdge instance:

1. In the OpenEdge Command Center console, click the **PASOE instances** tab.

The console displays a screen that lists all PAS for OpenEdge instances that are associated with all currently configured OpenEdge Command Center agents.

2. Select the check box for the PAS for OpenEdge instance for which you want to obtain process details.
3. In the **Actions** menu, click **Details**.

The Process Details window is displayed, showing the following information for each process that is running in the instance:

- Process type
- PID
- State
- CPU usage
- Memory usage
- Timestamp displaying when the process was started

Note: The Process Details window is not dynamically refreshed. To obtain the latest process details for the instance, click the refresh button.

Manage ABL applications, web applications, and REST services

The **ABL Applications** tab enables you to manage ABL applications deployed across multiple PAS for OpenEdge instances. The tab provides a consolidated view of ABL applications across multiple PAS for OpenEdge instances and the associated web applications and REST services.

This topic discusses how to manage ABL applications and their components using the **ABL Applications** tab.

View ABL applications across multiple PAS for OpenEdge instances

To view the ABL applications across multiple PAS for OpenEdge instances, click the **ABL Applications** tab in the OpenEdge Command Center console. The **ABL Applications** page appears, displaying a list of ABL applications and PAS for OpenEdge instances where the applications are deployed. By default, the first ABL application listed on this page is selected, and its details appear in the **Web Applications** and **Services** panes. You can view the details of any other listed ABL application by selecting it. The panes update automatically to display the details of the selected application.

You can sort the list by clicking the column header, and then clicking the arrow next to the header to toggle between ascending (A to Z) and descending (Z to A) order.

If the list of ABL applications is very long, you can filter them by entering an appropriate search string. You can filter the ABL applications based on their name, host name, and the PAS for OpenEdge instance they are deployed on.

The list of ABL applications is automatically refreshed periodically. However, to manually refresh the list, click the refresh icon.

From this page, you can deploy and undeploy ABL applications, web applications, and services. For more information, see [Manage ABL applications](#) on page 63.

If you click any ABL application, the **Edit ABL Application** page appears, where you can view, filter, and edit the properties of the corresponding ABL application. For more information, see [Edit ABL application](#) on page 68.

Web Applications pane

The **Web Applications** pane displays the web applications deployed on the ABL application selected on the **ABL Applications** page. The details of web applications are displayed in separate cards. Each card displays the following information about a web application:

- Name
- Service Count
- Path
- URI
- Secure URI

You can sort the web application cards by name or service counts. Click the **Refresh** button in the **Web Applications** pane to refresh the pane and the corresponding **Services** pane.

For more information about managing web applications, see [Manage web applications](#) on page 65.

Services pane

The **Services** pane displays the services deployed on the web application selected in the Web Applications pane. The details of services are displayed in separate cards. Each card displays the following information about a service:

- URI
- Secure URI
- Service Version
- Service Descriptor
- Service Location

You can sort the service cards by name. To refresh the content in the **Services** pane, click the refresh button.

Note: Currently, only REST services are displayed in the **Services** pane.

For more information about managing services, see [Manage REST services](#) on page 67.

Manage ABL applications

You can manage ABL applications from the **ABL Applications** page by performing the following tasks:

- Deploy ABL applications
- Undeploy ABL applications

Deploy ABL application

To deploy an ABL application on a PAS for OpenEdge instance:

1. In the OpenEdge Command Center console, click the **ABL Applications** tab.
The **ABL Applications** page appears, listing all ABL applications, web applications, and services deployed across multiple PAS for OpenEdge instances.
2. On the **ABL Applications** page, click **Actions > Deploy**. The **Deploy Application** dialog box appears.
3. In the **Select PASOE Instance** drop-down list, expand the agent node and then select the PAS for OpenEdge instance on which you want to deploy the ABL application.
4. Select the **ABL Application** option as the application type.
5. In the **Select Application File** field, click **Select File** to browse to the location of the OpenEdge Archive (OEAR) file containing the ABL application and select the file.

Note:

- You can deploy ABL applications as an OEAR file only. For information about the OEAR file, see "Create an OpenEdge Application Archive using tcman export" and "Create an OpenEdge Application Archive using an Ant Build" in *Manage Progress Application Server (PAS) for OpenEdge*.
 - If you are deploying an OEAR file on an instance that already contains ABL applications or web applications with the same names as those included in the OEAR file, you must undeploy the existing applications first.
 - The OEAR file does not need to have the same name as the ABL application.
 - You can upload an OEAR file only if its size is less than or equal to 523,239,424 bytes (499 MB). If you want to upload a larger file, update the `system-config.json` file by adding the `maxFileUploadSize` parameter with appropriate value and restart the OpenEdge Command Center server. The maximum value you can set is 3,221,225,472 bytes (3 GB). For more information, see "OpenEdge Command Center server configuration" in *Install the OpenEdge Command Center*.
-

6. Based on the status of the PAS for OpenEdge instance, the dialog box displays either the **Deploy** or **Deploy & Restart** button. The following table summarizes different scenarios:

If	Then	
Status of PAS for OpenEdge Instance	Deploy Application dialog box displays	Required action
Running	The Deploy & Restart button	Click Deploy & Restart .
Stopped	The Deploy button	Click Deploy .

Note: The deployment of the ABL application on the PAS for OpenEdge instance is independent of the deployment status of the Apache Tomcat Manager application.

A popup window appears, displaying the notification that the deploy operation is successfully submitted. You can click the bell icon to refresh and view the deployment status. After the successful deployment, the ABL application appears on the **ABL Applications** page.

Note: If an ABL application is deployed on a stopped PAS for OpenEdge instance, you must start the instance to access the ABL application services. For information on how to start a PAS for OpenEdge instance, see [Start or stop a PAS for OpenEdge instance](#) on page 60.

Undeploy ABL application

To undeploy an ABL application from a PAS for OpenEdge instance:

1. In the OpenEdge Command Center console, click the **ABL Applications** tab.

The **ABL Applications** page appears, listing all ABL applications, web applications, and services deployed across multiple PAS for OpenEdge instances.

2. Select the ABL application you want to undeploy, and click **Actions > Undeploy**. The **Undeploy ABL Application** dialog box appears.
3. Undeploy the ABL application. Based on the status of the PAS for OpenEdge instance and deployment of the Apache Tomcat Manager application, the dialog box displays either the **Undeploy** or **Undeploy & Restart** button. The following table summarizes different scenarios:

If		Then	
Status of PAS for OpenEdge instance	Status of Apache Tomcat Manager Application	Undeploy ABL Application dialog box displays	Required action
Running	Deployed	The Enter Apache Tomcat Manager credentials section and the Undeploy button	Enter the credentials and click Undeploy .
Running	Not deployed	The Undeploy & Restart button	Click Undeploy & Restart .
Stopped	NA	The Undeploy button	Click Undeploy .

A popup window appears, displaying the notification that the undeploy operation is successfully submitted. You can click the bell icon to refresh and view the undeployment status. After the successful undeployment, the ABL application no longer appears on the **ABL Applications** page.

For information about securing ABL applications, see [Secure online deployment of a new ABL application](#).

Manage web applications

You can manage web applications from the **ABL Applications** tab by performing the following tasks:

- Deploy web applications
- Undeploy web applications

Deploy web application

To deploy a web application into an ABL application on a PAS for OpenEdge instance:

1. In the OpenEdge Command Center console, click the **ABL Applications** tab.
The **ABL Applications** page appears, listing all ABL applications, web applications, and services deployed across multiple PAS for OpenEdge instances.
2. Click **Actions > Deploy**. The **Deploy Application** dialog box appears.
3. In the **Select PASOE Instance** drop-down list, expand the agent node and then select the PAS for OpenEdge instance on which you want to deploy the web application.
4. Select the **Web Application** option as the application type.
5. In the **Select ABL Application** field, type a new ABL application name or select the existing one from the drop-down list.

Note: If you type a new ABL application name, an ABL application with that name will be created and the web application will be deployed into it.

6. In the **Select Application File** field, click **Select File** to browse to the location of the WAR file containing the web application and select the file.

Note:

- If you are deploying a WAR file into an ABL application that already contains web applications with the same names as those included in the WAR file, you must undeploy the existing applications first.
- You can upload a WAR file only if its size is less than or equal to 523,239,424 bytes (499 MB). If you want to upload a larger file, update the `system-config.json` file by adding the `maxFileUploadSize` parameter with appropriate value and restart the OpenEdge Command Center server. The maximum value you can set is 3,221,225,472 bytes (3 GB).

For more information, see "OpenEdge Command Center server configuration" in *Install the OpenEdge Command Center*.

7. In the **Web Application Name** field, type the name of the web application.

Note: The web application name is automatically detected from the WAR file and is displayed in the **Web Application Name** field by default. You can change the application name if needed.

8. Based on the status of the PAS for OpenEdge instance and deployment of the Apache Tomcat Manager application, the dialog box displays either the **Deploy** or **Deploy & Restart** button. The following table summarizes different scenarios:

If		Then	
Status of PAS for OpenEdge instance	Status of Apache Tomcat Manager Application	Deploy Application dialog box displays	Required action
Running	Deployed	The Enter Apache Tomcat Manager credentials section and the Deploy button	Enter the credentials and click Deploy .
Running	Not deployed	The Deploy & Restart button	Click Deploy & Restart .
Stopped	NA	The Deploy button	Click Deploy .

Note: If you enter the incorrect Manager credentials, the application deployment becomes a normal deployment and requires the PAS for OpenEdge instance to restart.

A popup window appears, displaying the notification that the deploy operation is successfully submitted. You can click the bell icon to refresh and view the deployment status. After the successful deployment, the web application appears in the **Web Application** pane when you select the corresponding ABL application.

Undeploy web application

To undeploy a web application from an ABL application on a PAS for OpenEdge instance:

1. In the OpenEdge Command Center console, click the **ABL Applications** tab.

The **ABL Applications** page appears, listing all ABL applications, web applications, and services deployed across multiple PAS for OpenEdge instances.

2. Select the ABL application that contains the web application you want to undeploy. The associated web applications appear in the **Web Applications** pane.
3. Locate the card for the web application that you want to undeploy and then click **Undeploy** on the card. The **Undeploy Web Applications** dialog box appears.
4. Based on the status of the PAS for OpenEdge instance and deployment of the Apache Tomcat Manager application, the dialog box displays either the **Undeploy** or **Undeploy & Restart** button:

If		Then	
Status of PAS for OpenEdge instance	Status of Apache Tomcat Manager Application	Undeploy Web Applications dialog box displays	Required action
Running	Deployed	The Enter Apache Tomcat Manager credentials section and the Undeploy button	Enter the credentials and click Undeploy .
Running	Not deployed	The Undeploy & Restart button	Click Undeploy & Restart .
Stopped	NA	The Undeploy button	Click Undeploy .

A popup window appears, displaying the notification that the undeploy operation is successfully submitted. You can the bell icon to refresh and view the undeployment status. After the successful undeployment, the web application no longer appears in the **Web Applications** pane.

Manage REST services

From the **ABL Applications** tab, you can manage REST services in your web applications by performing the following tasks:

- Deploy services
- Undeploy services

Deploy services

To deploy a service in an existing web application:

1. In the OpenEdge Command Center console, click the **ABL Applications** tab.
The **ABL Applications** page is displayed.
2. On the **ABL Applications** page, select an ABL application.
The web applications and services deployed in the selected ABL application are displayed in the **Web Applications** and **Services** pane.
3. In the **Web Applications** pane, select the web application where you want to deploy a service.
4. In the **Services** pane, click **Deploy Service**.
The **Deploy Service** dialog box is displayed.
5. In the **Select service file** field, click **Select .paar**, browse to the location of the service file (. paar), and select the file.

Note: You can upload a PAAR file only if its size is less than or equal to 523, 239, 424 bytes (499 MB). If you want to upload a larger file, update the `system-config.json` file by adding the `maxFileUploadSize` parameter with appropriate value and restart the OpenEdge Command Center server. The maximum value you can set is 3, 221, 225, 472 bytes (3 GB).

For more information, see "OpenEdge Command Center server configuration" in *Install the OpenEdge Command Center*.

6. In the **Service name** field, type a name for the service.

Note: The name is automatically detected from the service (`.paar`) file and is displayed, by default. You can change the service name as desired.

7. Click **Deploy**.

A message is displayed confirming that the request to deploy the service is submitted.

8. Optional. Check notifications to view the status of service deployment.
9. After the service is deployed, it appears in the **Services** pane. If the associated web application or ABL application is stopped, you must start it to view the deployed service.

Undeploy services

To undeploy a service:

1. In the **Services** pane, locate the card of the service you want to undeploy.
2. In the service card, click **Undeploy**.

The **Undeploy Service** box is displayed.

3. Click **Undeploy**.

A message is displayed confirming that the request to undeploy a service is submitted.

4. Optional. Check notifications to confirm if the service was undeployed successfully.

Edit ABL application

This topic explains how you can filter and edit the properties of an ABL application.

On the **Edit ABL Application** page, you can view, filter, and edit the properties of an ABL application. All the properties are categorized into the following sections.

- MS Agent Startup Configurations
- MS Agent Session Procedures
- MS Agent Procedures
- MS Agent Logging Settings
- Session Manager Performance Tuning and Limit Settings
- Session Manager Timeouts

On this page, you can view a message at the top that displays the information on the current running status of the ABL application. You can use the search field to locate a specific property. You can also use the **Show all runtime updatable properties** toggle button to view the properties that you can modify without restarting the PAS for OpenEdge instance. However, this button is available only when the ABL application is running and enabled for runtime updates.

Filter ABL application properties

The **Edit ABL application** page displays all the properties of an ABL application. You can use a filter to easily locate the property you want to modify.

To filter the properties of the ABL application:

1. In the OpenEdge Command Center console, click the **ABL Applications** tab.
The **ABL Applications** page appears, listing all the ABL applications, web applications, and services deployed across multiple PAS for OpenEdge instances.
2. On the **ABL Applications** page, click the name of the ABL application.
The **Edit ABL Application** page appears, displaying the properties of the corresponding ABL application.
3. In the search field, enter the name, label, or section header of the property to locate it. You can also search the property using its name as specified in the `openedge.properties` file. For example, enter *workDir* (as in the `openedge.properties` file) to locate the **MS Agent Startup Directory** property.

Only the **MS Agent Startup Directory** property field is displayed on the **Edit ABL Application** page.

Edit ABL application properties

To edit the properties of an ABL application deployed on a PAS for OpenEdge instance:

1. In the OpenEdge Command Center console, click the **ABL Applications** tab.
The **ABL Applications** page appears, listing all the ABL applications, web applications, and services deployed across multiple PAS for OpenEdge instances.
2. On the **ABL Applications** page, click the name of the ABL application whose properties you want to modify. The **Edit ABL Application** page appears, displaying the properties of the corresponding ABL application. These properties are categorized into the following sections:
 - MS Agent Startup Configurations
 - MS Agent Session Procedures
 - MS Agent Procedures
 - MS Agent Logging Settings
 - Session Manager Performance Tuning and Limit Settings
 - Session Manager Timeouts
3. In the **MS Agent Startup Configurations** section, you can edit values for the following properties:

Field	Description
MS Agent Startup Directory	Specifies the path to the agent startup working directory.

Field	Description
MS Agent Startup Parameter	Specifies the server and agent startup parameters that each process uses when the broker starts it. For example, you can specify the Database (- db) parameter to connect to the application database. You can update this property dynamically. Dynamic changes affect only new MS Agents that are started after the update.
PROPATH	Specifies the PROPATH, which is a Progress environment variable, that contains a list of directories that the ABL Virtual Machine (AVM) uses to search for files and procedures to run. In this field, you can add, edit, delete, or reorder the PROPATH entries.

4. Click the **MS Agent Session Procedures** section to expand it, and then you can edit the values for the following properties:

Field	Description
Activate Procedure	Specifies the activate procedure for the session. This procedure must be a valid procedure on the PROPATH of the application service.
Deactivate Procedure	Specifies the deactivate procedure for the session. This procedure must be a valid procedure on the PROPATH of the application service.
Connect Procedure	Specifies the connect procedure for the session-managed sessions. This procedure must be a valid procedure on the PROPATH of the application service.
Disconnect Procedure	Specifies the disconnect procedure for session-managed sessions. This procedure must be a valid procedure on the PROPATH of the application service.
Session Startup Procedure	Specifies a startup procedure for the server. This procedure must be a valid procedure on the PROPATH of the application service.
Session Startup Procedure Parameter	Specifies the startup parameter that each session uses when the MS Agent starts it. You can update this property dynamically. Dynamic changes affect only new MS Agents that are started after the update.
Session Shutdown Procedure	Specifies the shutdown procedure for the server. This procedure must be a valid procedure on the PROPATH of the application service.

5. Click the **MS Agent Procedures** section to expand it, and then you can edit the values for the following properties:

Field	Description
MS Agent Startup Procedure	Specifies a startup procedure used for a multi-session agent. This procedure must be a valid procedure on the PROPATH of the application service.
MS Agent Startup Procedure Parameter	Specifies a character parameter that is provided as an input into the agentStartupProc procedure.
MS Agent Shutdown Procedure	Specifies a shutdown procedure that is used for a multi-session agent. This procedure must be a valid procedure on the PROPATH of the application service.

6. Click the **MS Agent Logging Settings** section to expand it, and then you can edit the values for the following properties:

Field	Description
Log File Name	Specifies the MS Agent log file that contains the location, name, and roll-over details of the MS Agent. If you want the MS Agent log file roll-over details at midnight, then add <code>{yyy-MM-dd}</code> . For example: <code><path-to>/<instance name>.agent.{yyyy-MM-dd}.log</code> .
Logging Level	Specifies the logging level for messages written into the MS Agent log buffer. The acceptable value range is 1 to 4.
Log Entry Types	Specifies a single entry or comma-delimited list of logging entry types for the MS Agent log buffer. By default, this parameter is set to <code>uBroker.basic</code> .

7. Click the **Session Manager Performance Tuning and Limit Settings** section to expand it, and then you can edit the values for the following properties:

Field	Description	Minimum value
Minimum Number of MS Agents	Specifies the minimum number of MS Agents expected to be active at any given time. If the number of MS Agents drops below this threshold, a client request triggers the session manager to start additional MS Agents to match this number.	1
Maximum Number of MS Agents	Specifies the maximum number of concurrent MS Agent operating the system processes that an individual ABL application can start on the AppServer.	1
Initial Number of MS Agents	Specifies the number of MS Agents to be started when the AppServer starts.	1

Field	Description	Minimum value
Maximum Connections per MS Agent	Specifies the maximum number of concurrent TCP connections allowed between the session manager and the MS Agent.	1
Maximum ABL Sessions per MS Agent	Specifies the maximum number of sessions an MS Agent can run.	1
Maximum Number of MS Agents to Start	Specifies the maximum number of MS Agents that can be started simultaneously. If this value is set to zero or not defined, no limit is enforced.	0

8. Click the **Session Manager Timeouts** section to expand it, and then you can edit the values for the following properties:

Field	Description	Minimum value
Request Wait Timeout	Specifies the maximum amount of time in milliseconds, for which a request waits in the session manager for a connection to become available before the request is rejected.	0
Connection Wait Timeout	Specifies the maximum amount of time in milliseconds, for which the session manager waits for a connection to be freed before creating a new connection.	0
Idle MS Agent Timeout	Specifies the timeout value in milliseconds, for an idle MS Agent. If an MS Agent is idle for longer than the specified timeout value, the session is deleted when the <code>idleResource</code> cleanup is done.	0
Idle Connection Timeout	Specifies the timeout value in milliseconds, between an AppServer client and the session manager. If a connection is idle for longer than the specified timeout value, the session manager terminates the connection by automatically disconnecting it from the AppServer.	0
Idle Resource Timeout	Specifies the timeout value, in milliseconds, to determine the frequency with which the PAS for OpenEdge server checks for idle resources. Any resource (for example, connection, MS Agent, or client session) that is not accessed for longer than the specified timeout for that property, will be terminated. If this property is set to zero, idle resource checking is disabled.	0

Field	Description	Minimum value
Idle Session Timeout	Specifies the timeout value in milliseconds for an idle session in the session manager. When using HTTP sessions, the APSV transport uses this value to set the expiration time of the HTTP session, which defaults to 30 minutes and is the same value that other transports use. The expiration of HTTP sessions is controlled in <code>web.xml</code> and does not require <code>idleResourceTimeout</code> to be set because it is managed by Tomcat.	0
MS Agent Listener Timeout	Specifies the maximum amount of time in milliseconds for which the session manager waits for an MS Agent to start, when the connection is established.	0

9. After editing the required properties, click **Apply**.

The **Confirm Changes** dialog box appears, displaying the details of the edited properties. The buttons in the dialog box vary depending on the running status of the ABL application and type of the properties edited:

- If the ABL application is running and you have edited:
 - Runtime updatable properties — the dialog box displays **Save** and **Cancel** buttons. Click **Save** to apply your changes.
 - Both types of properties or only the properties that require restart of the PAS for OpenEdge instance — the dialog box displays **Save**, **Save and Restart**, and **Cancel** buttons:
 - When you click **Save**, changes to the properties that do not require a restart of the PAS for OpenEdge instance are applied immediately, while changes to those properties requiring a restart will be applied after the PAS for OpenEdge instance restarts.
 - When you click **Save and Restart**. The **Confirm Restart** popup window appears. In the **Type restart to confirm** field, type *restart* and click **Restart** to restart the PAS for OpenEdge instance and apply changes to the properties. The ABL application will be unavailable during the restart.
- If the ABL application is not running, the dialog box displays only **Save** and **Cancel** buttons, irrespective of the type of properties edited. Click **Save** to save your changes. The changes to the properties will be applied after the ABL application is in running state.

Create users and manage roles using Authorization server

The Authorization server uses Role-Based Access Control (RBAC) to manage access to systems and their resources, including the OpenEdge Command Center server, the OpenEdge Command Center agent, and the Authorization server itself. It assigns permissions to specific roles instead of individual users, which simplifies the administration of access control. In addition to the predefined system roles, you can create custom roles that grant access to specific resources managed by the OpenEdge Command Center agent.

Using the Authorization server REST APIs, you can create users and assign or remove roles for a specific user, and create and manage custom roles in the Authorization server. This chapter covers tasks, such as logging in, creating users, assigning and removing roles. It also provides an overview of custom roles and answers frequently asked questions.

Note: The instructions in this chapter are written for Swagger UI, however, you can perform similar tasks using Postman.

Prerequisites

You must have the AUTHZ_ADMIN role to access the Authorization server and perform the tasks described in this chapter.

Log in to Authorization server

You can log in to the Authorization server as a user with the necessary privileges using both the OpenEdge Command Center server and Authorization server REST APIs.

To log in to the Authorization server:

1. Open a web browser and access the Swagger UI of the OpenEdge Command Center server by navigating to `https://<hostname>:<port>/openapi/`
2. Locate the **Authenticate your session** API with endpoint `POST - /api/auth/login` and expand it.
3. To enable the text area, click **Try it out**.
4. In **Request body**, type admin as username and password details in JSON format:

```
{
  "userName": "admin"
  "password": "<password>"
}
```

5. Click **Execute**. If authentication is successful, you see a JWT token in JSON format. For example:

```
{
  "idToken": "eyJhbGciOiJIUzI1NiJmZDNR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6ImFkbWluIiwiaWF0IjoxNTk1NDg2ODY3fQ.PZDjDXkQvpmSHWYXpt8JcYGeqEWX_8ehZonMY2aFHaa",
  "expiresIn": 240
}
```

6. Open another web browser and access the Swagger UI of the Authorization server by navigating to : `https://<hostname>:<port>/openapi-authz/`
7. Click **Authorize**.
8. In the **Available authorizations** dialog box, perform the following steps:
 - a) In the **Value** field, enter the JWT token that you obtained in Step 5.
 - b) Click **Authorize** and **Close**.

After successful authorization, you can create users, assign roles, and remove roles in the Authorization server.

Create a user

The Create a user API enables you to create a user in the Authorization server.

Note: Before you begin, ensure that you have created that specific user in the OpenEdge Command Center server.

To create a user in the Authorization server:

1. Log in to the Authorization server as a user with necessary privileges. For more information, see [Log in to Authorization server](#) on page 74.
2. Locate the Create a user API with endpoint POST - /authz/api/users and expand it.
3. To enable the text area, click **Try it out**.
4. In **Request body**, type the username in JSON format:

```
{
  "userName": "<Valid username>"
}
```

where, userName is the name of the user that you want to create.

5. Click **Execute**.

If the user creation is successful, you see the following success message.

```
{
  "message": "Request successful. The user is created in the Authorization Server."
}
```

Assign roles to a user

You can assign roles to users for granting them the required permission to access a system and perform specific tasks.

To assign roles to a user:

1. Log in to the Authorization server as a user with necessary privileges. For more information, see [Log in to Authorization server](#) on page 74.
2. Locate the Assign or remove roles API with endpoint PUT - /authz/api/users and expand it.
3. To enable the text area, click **Try it out**.
4. In the **Parameters** section, select assignRoles for the **action** parameter.
5. In **Request body**, provide user details and roles in JSON format, as follows:

```
{
  "userName": "string",
  "rolesToAssign": [{
    "roleURN": "string",
    "system": "string"
  }]
}
```

The following table describes the attributes in the request body:

Attribute name	Description
userName	Specifies the user to whom the roles need to be assigned.
roleURN	Identifies the roles within the Authorization server. It indicates what a specific role can access and where it is used. Its format includes the resource name it represents, role name, and, partition IDs (only when assigning roles to a user for an agent). The following are the possible values: • role:oecc/oecc_admin • role:authz/authz_admin • role:agent/<partitionid>/agent_admin • role:agent/<partitionid>/agent_resource_user • role:agent/<partitionid>/<custom-role-name> For more information about roleURN, see Roles in Authorization server on page 16. If you enter role:agent/<partitionid>/agent_admin or role:agent/<partitionid>/agent_resource_user as a roleURN value, you can get the agent partition ID using the Retrieve specific agent details API on the OpenEdge Command Center server. For more information, see "Retrieve specific agent details" in <i>OpenEdge Command Center REST API Reference</i>.

Attribute name	Description
system	Specifies the system on which the user with a specific role can access and perform tasks. The possible values are: - oecc - Indicates the OpenEdge Command Center server. - authz - Indicates the Authorization server. <partitionid> - Indicates the partition ID of the OpenEdge Command Center agent.

6. Click **Execute**.

If the roles assignment is successful, you see the response body in this format:

```
{
  "message": "Request successful. Roles are assigned to \"userName\"."
}
```

Example: Assign OECC_ADMIN and AUTHZ_ADMIN roles

This example demonstrates the request and response bodies for assigning the OECC_ADMIN and AUTHZ_ADMIN roles to the user named, James . Smith. This action grants the user access to the oecc (OpenEdge Command Center server) and authz (Authorization server) systems to do the following tasks:

System	Tasks
oecc	Manage the OpenEdge Command Center server, including viewing the details of server and database settings.
authz	Log in to the Authorization server and perform these tasks: - Create users - Manage roles

Request body

Following are the contents of the request body in JSON format:

```
{
  "userName": "James.Smith",
  "rolesToAssign": [
    {
      "roleURN": "role:oecc/oecc_admin",
      "system": "oecc"
    },
    {
      "roleURN": "role:authz/authz_admin",
      "system": "authz"
    }
  ]
}
```

Response body

Following are the contents of the response body in JSON format:

```
{
  "message": "Request successful. Roles are assigned to \"James.Smith\"."
}
```

Example: Assign AGENT_ADMIN and AGENT_RESOURCE_USER roles

This example demonstrates the request and response bodies for assigning the AGENT_ADMIN and AGENT_RESOURCE_USER roles to the user named, Sarah . Jones to access the oeccagent1 OpenEdge Command Center agent that has partition ID as db7253b679d9abc133904d4bf59d75f110c7023e57184d351be187f631d58ab8. This role assignment grants access of OpenEdge Command Center agent to the user, to do the following tasks:

- Perform actions on OpenEdge resources, including PAS for OpenEdge instances and OpenEdge databases that the agent manages.
- Log in to the Authorization server, create users, and assign the AGENT_ADMIN and AGENT_RESOURCE_USER roles to the user named Sarah . Jones .

Request body

Following are the contents of the request body in JSON format:

```
{
  "userName": "Sarah.Jones",
  "rolesToAssign": [{
    "roleURN":
"role:agent/db7253b679d9abc133904d4bf59d75f110c7023e57184d351be187f631d58ab8/agent_admin",
    "system": "db7253b679d9abc133904d4bf59d75f110c7023e57184d351be187f631d58ab8"
  }, {
    "roleURN":
"role:agent/db7253b679d9abc133904d4bf59d75f110c7023e57184d351be187f631d58ab8/agent_resource_user",
    "system": "db7253b679d9abc133904d4bf59d75f110c7023e57184d351be187f631d58ab8"
  }]
}
```

Response body

Following are the contents of the response body in JSON format:

```
{
  "message": "Request successful. Roles are assigned to \"Sarah.Jones\"."
}
```

Remove roles from a user

Remove roles from a user to ensure that they no longer have permissions to access the system.

To remove roles from a user:

1. Log in to the Authorization server as an admin user. For more information, see [Log in to Authorization server](#) on page 74.
2. Locate the Assign or remove roles API with endpoint PUT - /authz/api/users and expand it.
3. To enable the text area, click **Try it out** to enable the text area.
4. In the **Parameters** section, select removeRoles for the **action** parameter.
5. In **Request body**, provide user details and roles in JSON format.

```
{
  "userName": "string",
  "rolesToRemove": [
    "string"
  ]
}
```

The following table describes the attributes in the request body:

Attribute name	Description
userName	Specifies the user whose roles need to be removed.
roleURN	Identifies the role that needs to be removed from the user. For more information about roleURN, see Roles in Authorization server on page 16.

6. Click **Execute**. If the removal of roles is successful, you see the response body in this format:

```
{
  "message": "Request successful. These roles are removed for the user \"userName\": \"roleURN\""
}
```

Example

The following example demonstrates the request and response bodies for removing the AUTHZ_ADMIN role for the user named, JessicaWilson.

Request body

Following are the contents of the request body in JSON format:

```
{
  "userName": "JessicaWilson",
  "rolesToRemove": [
    "role:authz/authz_admin"
  ]
}
```

Response body

Following are the contents of the response body in JSON format:

```
{
  "message": "Request successful. These roles are removed for the user \"JessicaWilson\":  
  \"role:authz/authz_admin\"  
}
```

Create and manage custom roles

A custom role gives a user access to specific OpenEdge resources, such as a PAS for OpenEdge instance or an OpenEdge database that the OpenEdge Command Center agent manages. Unlike predefined system roles, which provide access to all the resources, a custom role applies only to the resources you choose.

When you grant access to a specific resource through a custom role, the user must also have access to all parent resources in the resource hierarchy. For example, to give a user access to a specific OpenEdge database, the user must also have access to the OpenEdge Command Center agent and the OpenEdge installation that host the database.

Before you create a custom role or add, modify, or remove permissions from an existing role, you must retrieve the resourceURN values for the target resources. For details, see "Retrieve resources for administered systems" in *Authorization Server REST API Reference*.

The Authorization server provides the following REST operations for creating and managing custom roles. For details about each operation, see the corresponding topic in *Authorization Server REST API Reference*.

- Create a custom role (POST /authz/api/roles)—Creates a custom role that grants permissions to selected resources.
- Update a custom role (PUT /authz/api/roles/{roleURN})—Adds, modifies, or removes permissions from a custom role.
- Delete custom roles (DELETE /authz/api/roles)—Deletes custom roles that are no longer required.

Note: Updating or deleting a custom role requires its roleURN. You can obtain this value from the response returned when the role is created or by retrieving it separately. For details, see "Retrieve roles for a system" in *Authorization Server REST API Reference*.

Frequently Asked Questions

This topic covers common scenarios for managing user access to OpenEdge Command Center agent and its OpenEdge resources.

How do I give a new user access to an existing OpenEdge Command Center agent to manage its OpenEdge resources?

Ensure that you meet the prerequisites, and then complete the following steps:

Prerequisites

- You have the AGENT_ADMIN role for the agent.
- The new user is already created in the OpenEdge Command Center server.
- You have obtained the agent partition ID for the agent using the `Retrieve specific agent details` API on the OpenEdge Command Center server. For more information, see "Retrieve specific agent details" in *OpenEdge Command Center REST API Reference*.

Steps

1. Log in to the Authorization server as a user with the required privileges. For more information, see [Log in to Authorization server](#) on page 74.
2. Create the user in the Authorization server. For more information, [Create a user](#) on page 75.
3. Assign the AGENT_RESOURCE_USER role to the user. For more information, see [Assign roles to a user](#) on page 75.

How do I give an existing user access to a new OpenEdge Command Center agent?

Ensure that you meet the prerequisites, and then complete the following steps:

Prerequisites

- You have the AGENT_ADMIN role for the agent.
- You have obtained the agent partition ID for the new OpenEdge Command Center agent using the `Retrieve specific agent details` API on the OpenEdge Command Center server. For more information, see "Retrieve specific agent details" in *OpenEdge Command Center REST API Reference*.

Steps

1. Log in to the Authorization server as a user with the required privileges. For more information, see [Log in to Authorization server](#) on page 74.
2. Assign the AGENT_ADMIN and AGENT_RESOURCE_USER roles to the user. For more information, see [Assign roles to a user](#) on page 75.

How do I give a user access to only a specific OpenEdge resource that the agent manages?

Ensure that you meet the prerequisites, and then complete the following steps:

Prerequisites

- You have the AUTHZ_ADMIN role.
- You have the AGENT_ADMIN role for the agent that manages the target resource.
- The user to whom you want to grant access already has access to the parent resources in the resource hierarchy, such as the OpenEdge Command Center agent and the OpenEdge installation that hosts the target resource. For example, if the target resource is an OpenEdge database, the user must have access to the OpenEdge Command Center agent and the OpenEdge installation that hosts the database.
- You have obtained the resourceURN for the target resource by using the `Retrieve resources for administered systems` API. For more information, see "Retrieve resources for administered systems" in the *Authorization server REST API reference*.

Steps

1. Log in to the Authorization server as a user with the required privileges. For more information, see [Log in to Authorization server](#) on page 74.
2. Create a custom role scoped to the target resource. For details, see [Create and manage custom roles](#) on page 80. The following request body defines a custom role with permissions for the target resource and its parent resources in the resource hierarchy.

```
{
  "roleName": "<role-name>",
  "definedForSystem": "<system-name>",
  "usedBySystem": "<system-name>",
  "description": "<role-description>",
  "permissions": [
    {
      "resourceURN": "<agent-resource-URN>",
      "operations": {
        "allowedOperations": [],
        "disallowedOperations": []
      }
    },
    {
      "resourceURN": "<installation-resource-URN>",
      "operations": {
        "allowedOperations": [],
        "disallowedOperations": []
      }
    },
    {
      "resourceURN": "<target-resource-URN>",
      "operations": {
        "allowedOperations": [],
        "disallowedOperations": []
      }
    }
  ]
}
```

Note: You can distribute the required resource URNs across multiple roles that you assign to the same user. However, the user must have access to all required parent resources across those roles.

3. Assign the custom role to the user. For more information, see [Assign roles to a user](#) on page 75.

Set up TLS for OpenEdge Command Center and MongoDB communication

You can enable TLS for secure communication between the OpenEdge Command Center server and MongoDB. You can configure the following types of authentication:

- Server authentication
- Mutual authentication

Server authentication

When using server authentication, the MongoDB server sends a certificate to the OpenEdge Command Center server to authenticate itself and ensure secure communication. To configure TLS server authentication:

1. In MongoDB installation, open the `bin/mongod.cfg` file in an editor.

Note: If the MongoDB installation is on the Linux platform, open the `etc/mongod.conf` file.

2. In the `network interface` section of the file, add the `tls` node.
3. In the `tls` node, add the following fields and enter the required values:

Field	Description
<code>mode</code>	Set value to <code>requireTLS</code> or <code>preferTLS</code> .
<code>certificateKeyFile</code>	Path of the public certificate of the MongoDB server that is signed by the Certificate Authority (CA).

4. Save your changes to the `bin/mongod.cfg` or `etc/mongod.conf` file and restart the MongoDB server.
5. In the OpenEdge Command Center server installation, open the `data/conf/db-config.json` file in an editor.
6. Add the `tls` field and set its value to `true`.
7. In `connectionOptions`, add the `sslCA` field.
8. For `sslCA`, enter the path of the public certificate of the CA that is used to validate the certificates presented by the OpenEdge Command Center server.
9. Save your changes to the `data/conf/db-config.json` file and restart the OpenEdge Command Center server.

After the OpenEdge Command Center server is started, the TLS handshake with the MongoDB server occurs and a secure channel is established.

Mutual authentication

When using mutual authentication, the OpenEdge Command Center server and the MongoDB server authenticate with each other before creating a secure communication channel. To configure TLS mutual authentication:

1. In MongoDB installation, open the `bin/mongod.cfg` file in an editor.

Note: If the MongoDB installation is on the Linux platform, open the `etc/mongod.conf` file.

2. In the `network interface` section of the file, add the `tls` node.
3. In the `tls` node, add the following fields and enter the required values:

Field	Description
<code>mode</code>	Set value to <code>requireTLS</code> or <code>preferTLS</code> .

Field	Description
certificateKeyFile	Path of the public certificate of the MongoDB server that is signed by the CA.
CAFile	Path of the file that contains the certificate chain for verifying the OpenEdge Command Center server's certificates.

4. Save your changes to the `bin/mongod.cfg` or `etc/mongod.conf` file and restart the MongoDB server.
5. In the OpenEdge Command Center server installation, open the `data/conf/db-config.json` file in an editor.
6. Add the `tls` field and set its value to `true`.
7. In `connectionOptions`, add the following fields and enter the required values:

Field	Description
sslCA	Path of the public certificate of the CA that is used to validate the certificates presented by the MongoDB server.
sslKey	The private key used for encryption.
sslCert	Path of the public certificate of the OpenEdge Command Center server that is signed by the CA.

8. Save your changes to the `data/conf/db-config.json` file and restart the OpenEdge Command Center server.

After the OpenEdge Command Center server is started, the TLS handshake with the MongoDB server occurs and a secure channel is established.

For more information about configuring MongoDB for TLS, see the following articles:

- <https://docs.mongodb.com/manual/tutorial/configure-ssl/>
- <https://docs.mongodb.com/manual/tutorial/configure-ssl-clients/>

Monitor OpenEdge resources using the OpenEdge Command Center agent

The OpenEdge Command Center agents can collect performance metrics of the following OpenEdge resources:

- OpenEdge database
- PAS for OpenEdge

The OpenEdge Command Center agent supports 12.2 and later versions of these OpenEdge resources.

Administrators can use these metrics to understand performance issues and accordingly tune the OpenEdge resources for optimal performance. The performance metrics are collected using the OpenTelemetry (OTel) standards. For more information about OpenTelemetry, see the [OpenTelemetry](#) documentation.

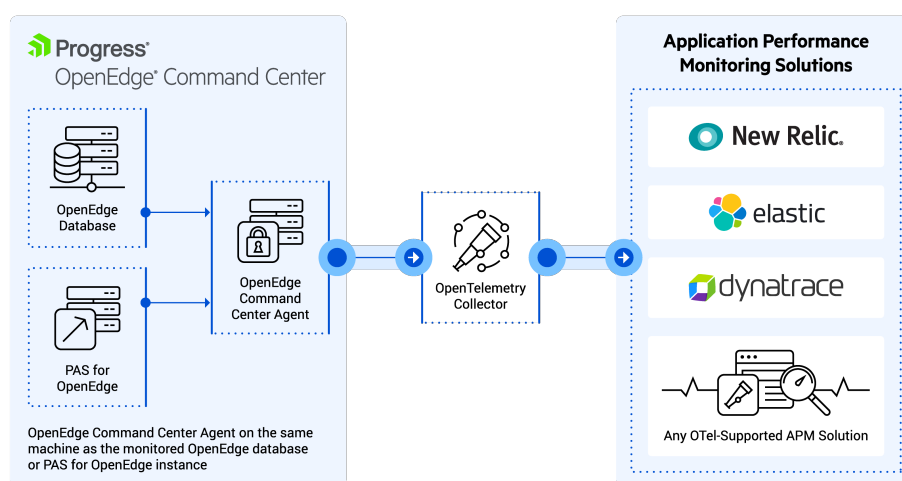
Since OTel is a vendor-agnostic and open-source technology, you can use any market-leading Application Performance Monitoring (APM) vendor solutions, such as Elastic APM, Dynatrace, NewRelic, and so on, to view the performance metrics.

Deployment architecture for monitoring OpenEdge resources

The components required to monitor OpenEdge resources are as follows:

- OpenEdge Command Center agent
- OTel Collector
- APM tool

The following diagram depicts the deployment architecture for monitoring OpenEdge resources using the OpenEdge Command Center agent:



OpenEdge Command Center agent—The OpenEdge Command Center agent is co-located with the OpenEdge resource you want to monitor. The agent monitors the OpenEdge resource, captures the required performance metrics, and then uses OpenTelemetry Protocol (OTLP) to transfer metrics data to OTel Collector.

OTel Collector—The OTel Collector is an application that processes telemetry data and sends it out to various destinations. To monitor the OpenEdge resources, the OTel Collector processes the performance metrics data collected by the OpenEdge Command Center agent and then exports it to the APM tool.

Application Performance Monitoring (APM) tool—APM tool is software that enables the observation and analysis of application performance. The tools may have data visualization capabilities that help administrators analyze performance and identify the bottlenecks. You can use market-leading APM tools that support the OTel standards, such as Elastic APM, Dynatrace, NewRelic, and so on, to view the performance metrics of OpenEdge resources. The APM tool uses the data exported by the OTel Collector.

Set up OpenTelemetry Collector

Set up OpenTelemetry (OTel) Collector to collect, process, and export performance metrics data of OpenEdge resources from the OpenEdge Command Center agent to various destinations, such as Application Performance Monitoring (APM) tool.

To set up OTel Collector, perform the following steps:

1. Download and install the appropriate version of OTel Collector for your platform. The latest certified version is 0.160.0. For more information about installing OTel Collector, see [OpenTelemetry Collector documentation](#).

Note: You can install OTel Collector on any system, but Progress recommends installing it on a different system from the one hosting the OpenEdge Command Center agent for optimal performance. You can also configure multiple agents to share performance metrics data with the same OTel Collector.

2. Open the `config.yaml` file of the OTel Collector installation directory in an editor.

The `config.yaml` file contains the following sections:

- `receivers`—Provide details about how OTel Collector can get data from the OpenEdge Command Center agent.
 - `processors`—Provide details about what OTel Collector does with the received data.
 - `exporters`—Provide details about where OTel Collector sends data for the Application Performance Monitoring (APM) tools.
3. In the `receivers` node, ensure that the value of the `endpoint` property is the same as specified for the `exporter.endpoint` property in the `otagentoedb.yaml` or `otagentpasoe.yaml` files.
 4. Edit the `config.yaml` file and provide information for the following properties in the `exporters` node:
 - Set `debug: verbosity` to `detailed`.
 - Set `file: path` to `./export.json`. This is the JSON file where OTel Collector saves metrics data for the APM tool.
 - Set `otlp/elastic: endpoint` to the location of the APM tool. For example, you may set the value to `localhost:8200`.
 5. Edit other properties listed in the `config.yaml` file per your requirements. To know more about these properties, see the [OpenTelemetry Collector documentation](#).
 6. Save the changes made to the `config.yaml` file.

Here is a sample `config.yaml` file:

```
receivers:
  otlp:
    protocols:
      grpc:
      http:
  otlp/withendpoint:
    protocols:
      grpc:
        endpoint: localhost:4317

exporters:
  debug:
    verbosity: detailed
  file:
    path: ./export.json
  otlp/elastic:
    endpoint: localhost:8200
    insecure: true

processors:
  batch:
  service:
    pipelines:
      traces:
        receivers: [otlp, otlp/withendpoint]
        exporters: [debug, otlp/elastic]
      metrics:
        receivers: [otlp, otlp/withendpoint]
        exporters: [debug, file, otlp/elastic]
```

7. Start OTel Collector after updating the `config.yaml` file by completing the following steps:

- a) Open a Command window and browse to the OTel Collector installation folder.
- b) Start OTel Collector by specifying the `config.yaml` file you updated. Use the following command:

```
<OTel Collector executable> --config config.yaml
```

For example on the Windows platform, use the following command:

```
otelcontribcol-0.129.1-windows_amd64.exe --config config.yaml
```

OpenTelemetry metrics for OpenEdge database

The OpenEdge Command Center agent can monitor an OpenEdge database and collect the following performance metrics.

The OpenEdge database metrics follow the naming convention `progress_oedb_<metric name>`. The prefix `progress_oedb` indicates that the metric is specific to the OpenEdge database.

Metric Name	Description
<code>summary_commits_total</code>	The number of transactions all users have committed.
<code>summary_undos_total</code>	The total number of transactions rolled back.
<code>record_updates_total</code>	The total number of records updated.

Metric Name	Description
record_reads_total	The total number of records read.
record_creates_total	The total number of records created.
record_deletes_total	The total number of records deleted.
summary_dbwrites_total	The total number of database blocks written to disk.
summary_dbreads_total	The total number of database blocks read.
summary_biwrites_total	The total number of Before-Image (BI) blocks written to disk.
summary_bireads_total	The total number of BI blocks read.
summary_aiwrites_total	The total number of After-Image (AI) blocks written to disk.
summary_recllocks_total	The total number of record locks used.
summary_recwaits_total	The total number of times users have waited to access a locked record.
summary_chkpts_total	The total number of checkpoints that have been performed.
summary_flushed_total	The total number of database buffers that have been flushed to disk because they were not written by the time the checkpoint ended.
recllock_waits_percent	The percentage of record accesses that resulted in a record lock wait, which occurs when the database engine must wait to access a locked record.
bibuf_waits_percent	The percentage of BI buffer waits, which occur when the database engine must wait to access a BI buffer.
aibuf_waits_percent	The percentage of AI buffer waits, which occur when the database engine must wait to access an AI buffer.
apw_writes_percent	The percentage of database blocks written to disk by the Asynchronous Page Writer (APW).
biw_writes_percent	The percentage of BI blocks written to disk by the Before-Image Writer (BIW).
aiw_writes_percent	The percentage of AI blocks written to disk by the After-Image Writer (AIW).
buffer_hits_percent	The percentage of buffer hits for both the primary and alternate buffer pools. A buffer hit occurs when the database engine locates a record in the buffer pool and does not have to read the record from the disk.
primary_hits_percent	The percentage of buffer hits for the primary buffer pool.

Metric Name	Description
alternate_hits_percent	The percentage of buffer hits for the alternate buffer pool.
instance_running_status	The operational status of an OpenEdge database. It uses numerical value 1 or -1 to represent the status. • 1 indicates that the database is running. • -1 indicates that the database is stopped.

Enable OpenEdge Command Center agent to collect performance metrics of OpenEdge database

You can enable the OpenEdge Command Center agent to collect performance metrics of an OpenEdge database by specifying details in the `otagentoedb.yaml` file. The file is present in the `conf` folder of the OpenEdge Command Center agent installation.

Follow these steps to enable the agent to collect performance metrics for an OpenEdge database:

1. Open the `otagentoedb.yaml` file in an editor.
2. In the `exporter` node of the `otagentoedb.yaml` file, provide values for the following properties of the OpenEdge Command Center agent connection to the OTel Collector:
 - `name`—A name for the OpenTelemetry exporter. Ensure that the value is set to `otlp`.
 - `endpoint`—The target URL to which the exporter sends the performance metrics data. The OTel Collector receives the metrics data at this endpoint. Make sure to use the same endpoint when you configure the OTel Collector.
 - `protocol`—The transport protocol used to export the metrics data. Use `grpc` as the value for this property.
 - `timeout`—The maximum time the OTLP exporter waits for each batch export.
 - `connectionretry`—The number of times the OpenEdge Command Center agent tries to connect to the OTel Collector in case of a connection failure. To know more about the impact of connection failure and how the connection is restored, see [Performance impact and resilience of collecting performance metrics](#) on page 102.
3. In the `oedbInstances` node, provide values for the following properties of the OpenEdge databases to be monitored:
 - `dbname`—The name of the database to be monitored.
 - `host`—The IP address of the database host.
 - `port`—The port on which the database is running.
 - `user`—The username of the database user.

Note: You must provide username of either a DBA user or a user with the SELECT permissions on the following virtual system tables (VSTs):

- `_ActSummary`
- `_ActRecord`
- `_ActPWs`
- `_ActBILog`
- `_ActAILog`
- `_ActBuffer`

-
- `password`—The password of the database user.

Note: The password can be in cleartext or encoded using the `genpassword` utility, which is a password encryption utility provided with the OpenEdge database.

- `metricsregex`—A regular expression (regex) to ensure that only the specified database performance metrics whose names match the pattern you have configured are collected. When left blank, the agent captures all the defined metrics.

Note: You can use only the `*` quantifier to create a regular expression.

- `otherdbconnparams`—Any other optional SQL JDBC connection parameters to be used for connecting to the database, separated by a semicolon (;).
- `dbschedule`—The time interval at which the agent must capture the metrics data. This property works in conjunction with the `dbduration` property. OpenTelemetry limits the frequency for posting data to a maximum of two times per minute or once every 30 seconds. Progress recommends to send metrics data once per minute.
- `dbduration`—The unit of time interval at which the agent must capture the metrics data. The possible values can be seconds, minutes, hours, or days. This property works in conjunction with the `dbschedule` property.

Note: You can provide details of multiple OpenEdge databases on your machine under the `oedbInstances` node and collect their metrics data using a single OpenEdge Command Center agent.

4. Save the changes made to the `otagentoedb.yaml` file.

Here is a sample `otagentoedb.yaml` file with details of two OpenEdge database instances:

```
exporter:
  name: "otlp"
  endpoint: "http://10.248.0.150:4317"
  protocol: "grpc"
  timeout: 10
  connectionretry: 20
```

```
oedbInstances:
- dbname: sports
  host: localhost
  port: 2022
  user: ss1
  password: ss1
  metricsregex:
  otherdbconnparams:
  dbschedule: 30
  dbduration: SECONDS
- dbname: testdb2
  host: localhost
  port: 3111
  user: admin
  password: xxx
  metricsregex:
  otherdbconnparams:
  dbschedule: 30
  dbduration: SECONDS
```

After updating the `otagentoedb.yaml` file, restart the OpenEdge Command Center agent.

Sample performance metrics data for OpenEdge database

When you configure OpenTelemetry (OTel) Collector to save the performance metrics data for an OpenEdge database in a JSON file and start the OTel Collector, the generated data is stored in the JSON file. You can choose to analyze this data directly or use an APM tool for detailed analysis.

The sample performance metrics data in the JSON file for the OpenEdge database is as follows:

```
{
  "resourceMetrics": [
    {
      "resource": {
        "attributes": [
          {
            "key": "agent_guid", "value": {"stringValue": "bdcd2d48-a600-46ce-89fd-9f9adc52a186"}},
          {
            "key": "hostname", "value": {"stringValue": "windowssrv2019"}},
          {
            "key": "port", "value": {"stringValue": "8686"}},
          {
            "key": "resource_name", "value": {"stringValue": "otoedb1"}},
          {
            "key": "service_name", "value": {"stringValue": "otoedb"}},
          {
            "key": "telemetry_sdk_language", "value": {"stringValue": "Java"}},
          {
            "key": "telemetry_sdk_name", "value": {"stringValue": "openTelemetry"}},
          {
            "key": "telemetry_sdk_os", "value": {"stringValue": "Windows"}},
          {
            "key": "telemetry_sdk_version", "value": {"stringValue": "1.15.0"}}
        ]
      },
      "scopeMetrics": [
        {
          "scope": {
            "name": "progress_oedb_otoedb1",
            "version": "1.0.0"
          },
          "metrics": [
            {
              "name": "progress_oedb_record_deletes_total",
              "description": "The number of records deleted",
              "unit": "cumulative",
              "sum": {
                "dataPoints": [
                  {
                    "startTimeUnixNano": "1718708278551640300",
                    "timeUnixNano": "1718708308566911000",
                    "asInt": "0"
                  }
                ],
                "aggregationTemporality": 2
              }
            },
            ...
          ]
        }
      ]
    }
  ]
}
```

Resource-level attributes

The resource-level attributes are key-value pairs that describe the characteristics of a monitored OpenEdge database including details about the OpenEdge Command Center agent, service name, and host name. These attributes remain constant across all collected metrics for that particular OpenEdge database and provide essential metadata for analysis and filtering metric data.

The following code snippet shows the resource-level attributes in the JSON file:

```
{
  "resourceMetrics": [
    {
      "resource": {
        "attributes": [
          {"key": "agent_guid", "value": {"stringValue": "bdcd2d48-a600-46ce-89fd-9f9adc52a186"}},
          {"key": "hostname", "value": {"stringValue": "windowssrv2019"}},
          {"key": "port", "value": {"stringValue": "8686"}},
          {"key": "resource_name", "value": {"stringValue": "otoedb1"}},
          {"key": "service_name", "value": {"stringValue": "otoedb"}},
          {"key": "telemetry_sdk_language", "value": {"stringValue": "Java"}},
          {"key": "telemetry_sdk_name", "value": {"stringValue": "openTelemetry"}},
          {"key": "telemetry_sdk_os", "value": {"stringValue": "Windows"}},
          {"key": "telemetry_sdk_version", "value": {"stringValue": "1.15.0"}}
        ]
      }
    }
  ]
}
```

The following table describes each resources-level attribute:

Attribute	Description
agent_guid	Specifies the unique identifier for the OpenEdge Command Center agent.
hostname	Specifies the name of the host system where the agent is running.
port	Specifies the port number on which the monitored OpenEdge database is running.
resource_name	Specifies the name of the OpenEdge database that is monitored.
service_name	Specifies the name of the telemetry service generating metric data.
telemetry_sdk_language	Specifies the programming language of the SDK used for metric data collection.
telemetry_sdk_name	Specifies the name of the telemetry SDK.
telemetry_sdk_os	Specifies the operating system of the host running the telemetry service.
telemetry_sdk_version	Specifies the version of the telemetry SDK used.

OpenTelemetry metrics for PAS for OpenEdge

The OpenEdge Command Center agent can monitor a PAS for OpenEdge instance and collect the following performance metrics.

The PAS for OpenEdge instance metrics follow the naming convention `progress_pasoe_<metric name>`. The prefix `progress_pasoe` indicates that the metric is specific to the PAS for OpenEdge instance.

Metrics Type	Metric Name	Description
REST transport	expression_errors_total	The total number of expression errors.
	failed_requests_total	The total number of failed requests.
	successful_run_requests_total	The total number of requests for which response was received successfully.
	successful_requests_total	The total number of requests successfully sent to the PAS for OpenEdge server.
	connect_requests_total	The total number of connection requests.
	status_requests_total	The total number of status type requests.
	requests_total	The total number of requests.
	successful_connect_requests_total	The total number of successful connection requests.
	service_unavailable_requests_total	The total number of requests for which services were not available.
	run_requests_total	The total number of run requests.
SOAP transport	url_notfound_errors_total	The total number of errors because of incorrectly supplied URLs.
	active_requests_total	The total number of requests that are in the ACTIVE state.
	wSDL_requests_total	The total number of WSDL requests.
	successful_requests_total	The total number of successful SOAP requests to the PAS for OpenEdge instance.
	method_notallowed_errors_total	The total number of errors caused because the requested method is not authorized.
	http_requests_errors_total	The total number of HTTP requests that resulted in errors.
	http_requests_total	The total number of HTTP requests.
	processor_errors_total	The total number of SOAP processor errors.

Metrics Type	Metric Name	Description
APSV transport	forbidden_requests_total	The total number of requests that failed with the 403 error code.
	disconnect_errors_total	The total number of disconnection errors.
	connect_errors_total	The total number of connection errors.
	disconnect_requests_total	The total number of disconnect requests.
	session_requests_total	The total number of session requests.
	session_errors_total	The total number of session errors.
WEB transport	head_requests_total	The total number of HEAD requests.
	trace_requests_total	The total number of TRACE requests.
	options_requests_total	The total number of OPTIONS requests.
	patch_requests_total	The total number of PATCH requests.
	get_requests_total	The total number of GET requests.
	servlet_requests_total	The total number of SERVLET requests.
	delete_requests_total	The total number of DELETE requests.
	put_requests_total	The total number of PUT requests.
	post_requests_total	The total number of POST requests.
	successful_servlet_requests_total	The total number of successful SERVLET requests.
	head_errors_total	The total number of HEAD requests that resulted in errors.
	trace_errors_total	The total number of TRACE requests that resulted in errors.

Metrics Type	Metric Name	Description
	options_errors_total	The total number of OPTIONS requests that resulted in errors.
	patch_errors_total	The total number of PATCH requests that resulted in errors.
	get_errors_total	The total number of GET requests that resulted in errors.
	delete_errors_total	The total number of DELETE requests that resulted in errors.
	put_errors_total	The total number of PUT requests that resulted in errors.
	post_errors_total	The total number of POST requests that resulted in errors.
	ablruntime_errors_total	The total number of ABL runtime errors.
	ablconnect_errors_total	The total number of ABL connection errors.
	failed_servlet_requests_total	The total number of SERVLET requests that resulted in errors.
Common	all_requests_total	The total number of requests to the PAS for OpenEdge instance.
	applications_total	The total number of applications deployed on the PAS for OpenEdge instance.
	agents_total	The total number of all agents spawned to execute the server requests.
	available_agents_total	The total number of agents with the AVAILABLE status.
	agent_sessions_total	The total number of sessions including all agents of the PAS for OpenEdge server.
	client_connections_total	The total number of client connections to the PAS for OpenEdge instance.
	client_sessions_total	The total number of client sessions connected to the PAS for OpenEdge instance.
	init_sessions_total	The total number of initial sessions configured for the PAS for OpenEdge instance.

Metrics Type	Metric Name	Description
	idle_sessions_total	The total number of sessions with the IDLE state.
	starting_sessions_total	The total number of starting sessions configured for the PAS for OpenEdge instance.
	available_sessions_total	The total number of sessions in the AVAILABLE state.
	reserved_sessions_total	The total number of sessions in the RESERVED state.
	stopped_sessions_total	The total number of sessions in the STOPPED state.
	instance_running_status	The operational status of a PAS for OpenEdge instance. It uses numerical value 1 or -1 to represent the status. • 1 indicates that the PAS for OpenEdge instance is running. • -1 indicates that the PAS for OpenEdge instance is stopped.

Enable OpenEdge Command Center agent to collect performance metrics of PAS for OpenEdge

You can enable the OpenEdge Command Center agent to collect performance metrics of a PAS for OpenEdge instance by specifying details in the `otagentpasoe.yaml` file. The file is present in the `conf` folder of the OpenEdge Command Center agent installation.

Follow these steps to enable the agent to collect performance metrics of a PAS for OpenEdge instance:

1. Open the `otagentpasoe.yaml` file in an editor.
2. In the `exporter` node of the `otagentpasoe.yaml` file, provide values for the following properties of the OpenEdge Command Center agent connection to the OTel Collector:
 - `name`—A name for the OpenTelemetry exporter. Ensure that the value is set to `otlp`.
 - `endpoint`—The target URL to which the exporter sends the performance metrics data. The OTel Collector receives the metrics data at this endpoint. Make sure to use the same endpoint when you configure the OTel Collector.
 - `protocol`—The transport protocol used to export the metrics data. Use `grpc` as the value for this property.
 - `timeout`—The maximum time the OTLP exporter waits for each batch export.
 - `connectionretry`—The number of times the OpenEdge Command Center agent tries to connect to the OTel Collector in case of a connection failure. To know more about the impact of connection failure and how the connection is restored, see [Performance impact and resilience of collecting performance metrics](#) on page 102.
3. Under the `pasInstances` node of the YAML file, provide values for the following properties of the PAS for OpenEdge instances to be monitored:
 - `pasdir`—The location of the PAS for OpenEdge instance.

Note: Make sure that a PAS for OpenEdge already exists at the specified location before you provide the value for the `pasdir` property.

- `passchedule`—The time interval at which the agent must capture the metrics data. This property works in conjunction with the `pasduration` property. OpenTelemetry limits the frequency for posting data to a maximum of two times per minute or once every 30 seconds. Progress recommends sending metrics data once per minute.
- `pasduration`—The unit of time interval at which the agent must capture the metrics data. The possible values can be seconds, minutes, hours, or days. This property works in conjunction with the `passchedule` property.
- `metricsregex`—A regular expression (regex) to ensure that only the specified PAS for OpenEdge instance performance metrics whose names match the pattern you have configured are collected. When left blank, the agent captures all the defined metrics.

Note: You can use only the `*` quantifier to create a regular expression.

4. Save the changes made to the `otagentpasoe.yaml` file.

Here is a sample `otagentpasoe.yaml` file with details of two PAS for OpenEdge instances:

```
exporter:
  name: "otlp"
  endpoint: "http://localhost:4317"
  protocol: "grpc"
  timeout: 10
  connectionretry: 20

pasInstances:
  - pasdir: "C:/OpenEdge/WRK/oepas5"
```

```
passchedule: 30
pasduration: SECONDS
metricsregex:
- pasdir: "C:/OpenEdge/WRK/oepas2"
  passchedule: 30
  pasduration: SECONDS
  metricsregex:
```

After updating the `otagentpasoe.yaml` file, restart the OpenEdge Command Center agent.

Sample performance metrics data for PAS for OpenEdge

When you configure OpenTelemetry (OTel) Collector to save the performance metrics data for a PAS for OpenEdge instance in a JSON file and start the OTel Collector, the generated data is stored in the JSON file. You can choose to analyze this data directly or use an APM tool for detailed analysis.

The sample performance metrics data in the JSON file for the PAS for OpenEdge instance is as follows:

```
{
  "resourceMetrics": [
    {
      "resource": {
        "attributes": [
          {"key": "agent_guid", "value": {"stringValue": "ABCD"}},
          {"key": "hostname", "value": {"stringValue": "windowssrv2019"}},
          {"key": "resource_name", "value": {"stringValue": "RetailMgmt"}},
          {"key": "service_name", "value": {"stringValue": "otpasoe"}},
          {"key": "telemetry_sdk_language", "value": {"stringValue": "Java"}},
          {"key": "telemetry_sdk_name", "value": {"stringValue": "openTelemetry"}},
          {"key": "telemetry_sdk_os", "value": {"stringValue": "Windows"}},
          {"key": "telemetry_sdk_version", "value": {"stringValue": "1.15.0"}}
        ]
      },
      "scopeMetrics": [
        {
          "scope": {
            "name": "progress_oepas_C:/OE_128/wrk/RetailMgmt",
            "version": "1.0.0"
          },
          "metrics": [
            {
              "name": "Progress_pasoe_get_errors_total",
              "description": "The number of GET requests that resulted in errors.",
              "unit": "cumulative",
              "sum": {
                "dataPoints": [
                  {
                    "attributes": [
                      {"key": "ablapp", "value": {"stringValue": "RetailMgmt"}},
                      {"key": "transport", "value": {"stringValue": "WEB"}},
                      {"key": "webapp", "value": {"stringValue": "ROOT"}}
                    ],
                    "startTimeUnixNano": "1718893655587297400",
                    "timeUnixNano": "1718893685609868500",
                    "asInt": "0"
                  },
                  {
                    "attributes": [
                      {"key": "ablapp", "value": {"stringValue": "Customer"}},
                      {"key": "transport", "value": {"stringValue": "WEB"}},
                      {"key": "webapp", "value": {"stringValue": "Orders"}}
                    ],
                    "startTimeUnixNano": "1718893655587297400",
                    "timeUnixNano": "1718893685609868500",
                    "asInt": "0"
                  }
                ]
              },
              "aggregationTemporality": 2
            },
            ...
          ]
        }
      ]
    }
  ]
}
```

Resource-level attributes

The resource-level attributes are key-value pairs that describe the characteristics of a monitored PAS for OpenEdge instance, including details about the OpenEdge Command Center agent, service name, and host name. These attributes remain constant across all collected metrics for that particular PAS for OpenEdge instance and provide essential metadata for analysis and filtering metric data.

The following code snippet shows the resource-level attribute in the JSON file:

```
{
  "resourceMetrics": [
    {
      "resource": {
        "attributes": [
          {"key": "agent_guid", "value": {"stringValue": "ABCD"}},
          {"key": "hostname", "value": {"stringValue": "windowssrv2019"}},
          {"key": "resource_name", "value": {"stringValue": "RetailMgmt"}},
          {"key": "service_name", "value": {"stringValue": "otpasoe"}},
          {"key": "telemetry_sdk_language", "value": {"stringValue": "Java"}},
          {"key": "telemetry_sdk_name", "value": {"stringValue": "openTelemetry"}},
          {"key": "telemetry_sdk_os", "value": {"stringValue": "Windows"}},
          {"key": "telemetry_sdk_version", "value": {"stringValue": "1.15.0"}}
        ]
      }
    }
  ]
  ...
}
```

The following table describes each resources-level attribute:

Attribute	Description
agent_guid	Specifies the unique identifier for the OpenEdge Command Center agent.
hostname	Specifies the name of the host system where the agent is running.
resource_name	Specifies the name of the PAS for OpenEdge instance that is monitored.
service_name	Specifies the name of the telemetry service generating metric data.
telemetry_sdk_language	Specifies the programming language of the SDK used for metric data collection.
telemetry_sdk_name	Specifies the name of the telemetry SDK.
telemetry_sdk_os	Specifies the operating system of the host running the telemetry service.
telemetry_sdk_version	Specifies the version of the telemetry SDK used.

Metric-level attributes

The metric-level attributes are key-value pairs within the data point of a recorded metric. They provide additional contextual information about the measurement being taken. These attributes are dynamic and allow you to view, filter, and perform granular analysis of metric data.

The following code snippet shows the metric-level attributes in the JSON file:

```
...
"metrics": [
  {
    "name": "Progress_pasoe_get_errors_total",
    "description": "The number of GET requests that resulted in errors.",
    "unit": "cumulative",
    "sum": {
      "dataPoints": [
        {
          "attributes": [
            {"key": "ablapp", "value": {"stringValue": "RetailMgmt"}},
            {"key": "transport", "value": {"stringValue": "WEB"}},
            {"key": "webapp", "value": {"stringValue": "ROOT"}}
          ],
          "startTimeUnixNano": "1718893655587297400",
          "timeUnixNano": "1718893685609868500",
          "asInt": "0"
        },
        {
          "attributes": [
            {"key": "ablapp", "value": {"stringValue": "Customer"}},
            {"key": "transport", "value": {"stringValue": "WEB"}},
            {"key": "webapp", "value": {"stringValue": "Orders"}}
          ],
          "startTimeUnixNano": "1718893655587297400",
          "timeUnixNano": "1718893685609868500",
          "asInt": "0"
        }
      ],
      "aggregationTemporality": 2
    }
  },
  ...
]
```

Performance impact and resilience of collecting performance metrics

The OpenEdge Command Center agent runs on the same machine as the monitored OpenEdge resource and regularly interacts with the resource for collecting the performance metrics data. However, this action has a negligible impact on the performance of the OpenEdge resource.

Installing OTel Collector on the same machine as the OpenEdge Command Center agent and the monitored OpenEdge resource can impact the performance of the OpenEdge resource. So, it is recommended to install OTel Collector on a different machine.

You can further reduce the performance impact by increasing the time interval at which the agent scrapes the metrics data. To increase the time interval:

- When monitoring an OpenEdge database, increase the value of the `dbschedule` property in the `otagentodb.yaml` file.
- When monitoring a PAS for OpenEdge instance, increase the value of the `paschedule` property in the `otagentpasoe.yaml` file.

Contact Progress Technical Support if you have any queries on the performance impact.

When the monitored OpenEdge resources are running, the OpenEdge Command Center agent continues to collect metrics data and sends it to the OTel Collector. The collection of metrics data ceases when the OpenEdge Command Center agent stops. If the monitored OpenEdge resource stops and then restarts, the collection of metrics data resumes if both the OpenEdge Command Center agent and OTel Collector are running.

If the OTel Collector stops or the connection between the OpenEdge Command Center agent and the OTel Collector is disrupted, the Command Center agent tries to reconnect as per the value configured for the `connectionretry` property in the `otagentoedb.yaml` or `otagentpasoe.yaml` file.

- If the connection is restored within the configured retry attempts, the Command Center agent sends the performance data to the OTel Collector.
- If the connection is not restored within the configured retry attempts, the Command Center agent stops sending the performance data to the OTel Collector and ceases collecting the data from the OpenEdge resource. You must restart both the OpenEdge Command Center agent and the OTel Collector to resume monitoring of the OpenEdge resource.

If the connection between the OpenEdge Command Center agent and the OTel Collector is disrupted due to an incorrectly configured YAML file on the Command Center agent, update the configuration file and restart the Command Center agent.

Reset super admin user details

Note: The super admin account is a local account. In IdP authentication mode, local accounts cannot sign in, so resetting the super admin account has no effect while an identity provider configuration is active. If you must recover local administrative access, activate the local configuration with the `SWITCHAUTHMODE` utility, restart the OpenEdge Command Center server, and then run this utility.

In local authentication mode, you can modify the following details of the super admin user after launching the OpenEdge Command Center server:

- First name
- Last name
- Password
- Email address
- Description

To modify the super admin user details, perform the following steps:

1. Open a command-line interface with administrative privileges. On Windows, run the command prompt as an administrator. On Linux, run the terminal as a root user.
2. Change to the `utils` directory within the directory where the OpenEdge Command Center server is installed.
 - On Windows, go to `cd C:\Progress\OECC\Server\utils`.
 - On Linux, go to `cd /Progress/OECC/server/utils`.
3. To modify one or more details of a super admin user, run the following command:

```
resetsuperadmin [--firstname <first name>] | [--lastname <last name>] | [--pwd <password>] | [--email <email>] | [--description <description>]
```

Replace each placeholder with the appropriate value. For detailed information about the syntax and parameters, see [RESETSUPERADMIN utility](#) on page 127.

Troubleshoot identity provider authentication

To resolve sign-in and authorization problems in IdP authentication mode, use the following information:

Message	Cause	Resolution
Your identity provider did not return an email address. Contact your IdP Administrator.	The identity provider did not supply an email address for the user.	In the identity provider, include the email scope and ensure that the email claim is issued for the OpenEdge Command Center application.
User does not have API access / Error while fetching recently discovered resources.	The user exists in the identity provider but is not assigned any OpenEdge Command Center app role.	Assign an app role to the user, or a group that contains the user. See Required IdP app roles for OpenEdge Command Center .
Access denied.	The app role value in the access token does not match any role that is defined in OpenEdge Command Center.	Correct the app role value in the identity provider so that it matches the OpenEdge Command Center role value exactly, including case.
Authentication service unavailable. Try again later.	The identity provider is unreachable, or the OIDC discovery document cannot be retrieved from the issuer URL.	1. Verify network connectivity from the OpenEdge Command Center server to the identity provider. 2. Verify that the issuer URL serves a valid discovery document.
Authentication failed. Please try again.	The sign-in could not be verified, usually because it was started from a browser tab that is no longer current, or because it was not completed in time.	1. Start the sign-in again from the OpenEdge Command Center login page. 2. Confirm that browser cookies are enabled.
Authentication configuration error. Contact system admin.	The active authentication configuration is incomplete or invalid.	1. Review the stored configuration with <code>SWITCHAUTHMODE --LIST</code>. 2. Correct the values, activate the configuration, and restart the server.

Consider the following additional conditions:

- **Roles that were changed in the identity provider are not applied**—A role change takes effect when OpenEdge Command Center next renews the access of the user, or at the next sign-in of that user.
- **Users are prompted to sign in again about every hour**—The `offline_access` scope is missing, so the identity provider does not issue a refresh token. Add the scope to the configuration.

- **Roles cannot be read from the access token**—The identity provider issued an opaque access token instead of a JWT.
 - For Microsoft Entra ID, expose the custom API scope, set the requested access token version to 2, and do not request Microsoft Graph scopes in the same request.
 - For Okta, use a custom authorization server.

Note: In IdP authentication mode, POST /api/auth/login and POST /api/auth/refresh_token are blocked. These endpoints are not a future gap.

Learn about the OpenEdge Command Center MCP Server

The OpenEdge Command Center MCP Server lets MCP-compatible AI clients, such as VS Code with Copilot and Claude Desktop, retrieve authorized OpenEdge Command Center information through read-only tools and prompts.

The OpenEdge Command Center MCP Server is included with the OpenEdge Command Center Server installer and is installed as part of the server installation. No separate MCP Server installation is required. It provides access to information such as Server nodes, Agents, OpenEdge installations, PAS for OpenEdge instances, applications, and list databases.

OpenEdge Command Center Server remains responsible for authorization. The tools and prompts return only information that the authenticated user is authorized to access according to the user's OpenEdge Command Center roles and permissions. The MCP server does not grant additional permissions or provide tools that modify OpenEdge Command Center resources.

For details, see the following topics:

- [Learn how the OpenEdge Command Center MCP Server works](#)
- [Explore tools and prompts](#)
- [Understand the OpenEdge Command Center MCP Server prerequisites](#)
- [Learn about authentication and authorization](#)
- [Configure and manage the OpenEdge Command Center MCP Server](#)
- [Start, stop, and check the OpenEdge Command Center MCP Server](#)
- [Verify the OpenEdge Command Center MCP Server](#)

- [Connect to the OpenEdge Command Center MCP Server](#)
- [Troubleshoot OpenEdge Command Center MCP Server connections](#)

Learn how the OpenEdge Command Center MCP Server works

The OpenEdge Command Center MCP Server lets an MCP-compatible AI client retrieve authorized information from OpenEdge Command Center by using read-only tools and prompts.

The MCP Server runs as a separate process and communicates with MCP clients over HTTPS through the /mcp endpoint. The MCP server requests information from the OpenEdge Command Center Server REST API and returns the results to the AI client.

Each client connection is authenticated separately. The authentication method depends on the mode configured for OpenEdge Command Center Server:

- **Local authentication:** In this authentication mode, you create a personal access token in OpenEdge Command Center and use it to connect the MCP client.
- **IdP authentication:** In this, you configure an external identity provider for OpenEdge Command Center. Then use the credential issued by the configured identity provider to connect the MCP client. For more information, see [Learn about external identity provider \(IdP\) authentication](#).

OpenEdge Command Center Server remains responsible for authorization. The MCP server does not provide additional permissions. The tools and prompts return only the information that the authenticated user is authorized to access.

Explore tools and prompts

The OpenEdge Command Center MCP Server provides read-only tools and prompts for retrieving authorized information from OpenEdge Command Center.

- A direct tool retrieves one type of information, such as Server nodes, Agents, PAS instances, Web applications, and transports.
- A composite tool combines information from multiple direct tools to provide a broader, task-oriented result. For example, a composite tool can combine PAS details, process information, filesystem-size information, properties, and related notifications into one report.
- A prompt guides the MCP client to use read-only tools and present the results.
- The MCP Server determines the information returned by its tools, subject to the authenticated user's OpenEdge Command Center permissions. How the MCP client and its underlying LLM interpret, summarize, or present that information may vary.

The information returned by tools and prompts depends on the authenticated user's OpenEdge Command Center roles and permissions. OpenEdge Command Center Server remains responsible for authorization, and the MCP server does not grant additional permissions.

Tools and prompts do not create, update, delete, deploy, start, stop, or otherwise modify OpenEdge Command Center resources. They do not manage credentials, passwords, personal access tokens, or multifactor authentication.

Tools

OpenEdge Command Center Server and identity

Tool	Inputs	Description
oecc_server_list	None	Returns the OpenEdge Command Center Server nodes visible to the authenticated user, including available node information.
oecc_server_about	None	Returns the OpenEdge Command Center Server product, version, and build information.
oecc_user_get_current	None	Returns the authenticated user's OpenEdge Command Center identity and assigned roles.

Agents and OpenEdge installations

Tool	Inputs	Description
oecc_agent_list	filter (optional)	Returns authorized OpenEdge Command Center Agents and their available inventory and connection information. Use a filter to narrow the results.
oecc_agent_get	agentId (required)	Returns details of the Agent identified by its Agent resource key, including available identity and host information.
oecc_oe_install_list	None	Returns OpenEdge installations discovered by Agents that the authenticated user is authorized to access.

PAS for OpenEdge instances

Tool	Inputs	Description
oecc_pas_list	filter (optional)	Returns authorized PAS for OpenEdge instances and their available inventory and status information. Use a filter to narrow the results.
oecc_pas_get	resKey (required)	Returns the recorded details of a PAS for OpenEdge instance, including available identity, host, status, and configuration information.

Tool	Inputs	Description
oecc_pas_get_properties	resKey (required)	Returns the non-secret configuration properties of a PAS for OpenEdge instance.
oecc_pas_get_process_details	resKey (required)	Returns available process details for a PAS for OpenEdge instance, including process identifiers, CPU use, memory use, and uptime.
oecc_pas_get_size	resKey (required)	Returns available filesystem and size information for a PAS for OpenEdge instance.

Applications

Tool	Inputs	Description
oecc_ablapp_list	filter (optional)	Returns deployed ABL applications visible to the authenticated user, including available PAS relationships and status information.
oecc_ablapp_get_properties	pasResKey, ablAppName (required)	Returns configuration properties for the specified ABL application under a PAS for OpenEdge instance.
oecc_webapp_list	filter (optional)	Returns deployed web applications and their available parent-application information.

Databases

Tool	Inputs	Description
oecc_db_list	filter (optional)	Returns authorized OpenEdge databases and their available inventory and status information.
oecc_db_get	resKey (required)	Returns the recorded details of an OpenEdge database, including available identity, host, and connection-state information.

These tools do not provide database schema, table, or storage drill-down.

Metadata and operations

Tool	Inputs	Description
oecc_metadata_labels	None	Returns labels associated with authorized OpenEdge Command Center resources.

Tool	Inputs	Description
oecc_metadata_fields	entity (required)	Returns filterable field names and observed values for an agent, pas, ablapp, or database entity.
oecc_operation_get	operationId (required)	Returns the current status and available result of an authorized OpenEdge Command Center operation. It does not initiate or modify the operation.

For `oecc_metadata_fields`, use the returned `fieldName` value as the `field` value in a filter. Field discovery is not available for web applications or notifications.

APSV transport

All APSV transport tools require `pasResKey`, `ablAppName`, and `webAppName`.

Tool	Inputs	Description
oecc_apsv_metrics_get	pasResKey, ablAppName, webAppName (required)	Returns available runtime metrics for the APSV transport of the specified web application.
oecc_apsv_configuration_get	pasResKey, ablAppName, webAppName (required)	Returns the configured APSV transport properties of the specified web application.

REST transport

All REST transport tools require `pasResKey`, `ablAppName`, and `webAppName`.

Tool	Inputs	Description
oecc_restsvc_list	pasResKey, ablAppName, webAppName (required)	Returns REST services deployed under the specified web application.
oecc_rest_metrics_get	pasResKey, ablAppName, webAppName (required)	Returns available runtime metrics for the REST transport of the specified web application.
oecc_rest_configuration_get	pasResKey, ablAppName, webAppName (required)	Returns the persistent REST transport configuration of the specified web application.
oecc_rest_runtime_properties_get	pasResKey, ablAppName, webAppName (required)	Returns the current, non-persisted REST transport properties of the specified web application.

SOAP transport

All SOAP transport tools require `pasResKey`, `ablAppName`, and `webAppName`.

Tool	Inputs	Description
oecc_soap_metrics_get	pasResKey, ablAppName, webAppName (required)	Returns available runtime metrics for the SOAP transport of the specified web application.
oecc_soap_configuration_get	pasResKey, ablAppName, webAppName (required)	Returns the persistent SOAP transport configuration of the specified web application.
oecc_soap_services_list	pasResKey, ablAppName, webAppName (required)	Returns SOAP services deployed under the specified web application.
oecc_soap_runtime_properties_get	pasResKey, ablAppName, webAppName (required)	Returns the current, non-persisted SOAP transport properties of the specified web application.

Web transport

All Web transport tools require pasResKey, ablAppName, and webAppName.

Tool	Inputs	Description
oecc_web_metrics_get	pasResKey, ablAppName, webAppName (required)	Returns available runtime metrics for the Web transport of the specified web application.
oecc_web_configuration_get	pasResKey, ablAppName, webAppName (required)	Returns the persistent Web transport configuration of the specified web application.
oecc_web_handlers_list	pasResKey, ablAppName, webAppName (required)	Returns handlers configured for the Web transport of the specified web application.

Composite tools

Tool	Inputs	Description
oecc_environment_overview	include, filter (optional)	Combines resource counts, status breakdowns, notable unhealthy resources, and OpenEdge Command Center Server node information into an environment overview. The optional include argument accepts servers, agents, installs, pas, ablapps, webapps, and databases; all are included by default.

Tool	Inputs	Description
oecc_pas_diagnostics	pasResKey (required), includeProperties (optional, default false)	Combines a PAS for OpenEdge instance's identity and status with available process and disk information. Include non-secret properties when includeProperties is enabled.
oecc_webapp_transport_health	pasResKey, ablAppName, webAppName (required), transports (optional)	Combines available metrics and configuration information for the specified web application's REST, Web, APSV, and SOAP transports. All transports are checked by default.

If an optional source fails, a composite tool preserves successfully retrieved information and reports the failed source in `partialFailures`.

Prompts

All prompt arguments are optional. Prompts guide the MCP client to call read-only tools and support its conclusions with retrieved tool output. Prompts do not query OpenEdge Command Center independently.

Prompt	Arguments	Description
oecc_pasoe_health_check	filter	Guides the model to list PAS for OpenEdge instances, retrieve each selected instance and its process details, summarize overall health, and identify stopped or abnormal instances with supporting evidence and investigation guidance.
oecc_pasoe_memory_analysis	filter, thresholdPercent (default 80)	Guides the model to correlate PAS process and size information, flag usage above the specified threshold, and summarize memory health across the selected environment.
oecc_pasoe_incident_triage	pasResKey, ablAppName, webAppName	Guides the model to review PAS, process, Agent, and available transport information to identify a likely incident cause. When all three application identifiers are supplied, transport analysis focuses on that web application.
oecc_pasoe_transport_status	pasResKey, ablAppName, webAppName	Guides the model to discover the application scope, inspect REST, Web, APSV, and SOAP configuration, list services or handlers, and summarize which transports appear configured, enabled, and healthy.

Prompt	Arguments	Description
oecc_database_health_check	filter	Guides the model to list databases, inspect each selected database, summarize availability and connection health, and identify unavailable resources. Storage- and table-level analysis is not included.
oecc_application_inventory	filter	Guides the model to correlate PAS for OpenEdge instances, ABL applications, web applications, application properties, and exposed services or handlers into an inventory grouped by PAS instance.
oecc_environment_overview_report	None	Guides the model to retrieve OpenEdge Command Center release information and inventory for Server nodes, OpenEdge installations, PAS for OpenEdge instances, databases, and Agents, then summarize the environment composition and health.

Prompts have no independent audit event because they do not call OpenEdge Command Center. Each tool invocation initiated from a prompt follows the normal authentication and auditing flow.

Understand the OpenEdge Command Center MCP Server prerequisites

The OpenEdge Command Center MCP server runs as a separate process and communicates with MCP clients over HTTPS. By default, it listens on port 8002. The OpenEdge Command Center server user API listens on port 8000 by default.

Before connecting, confirm that:

- The MCP client can reach the MCP Server.
- The MCP Server can reach the OpenEdge Command Center Server User API.
- The MCP Server is running.
- The MCP client trusts the MCP Server certificate.

To install the default root CA or configure certificates for remote access and information about `mcpConfig.json` settings, see [Configure and manage the OpenEdge Command Center MCP Server](#).

Learn about authentication and authorization

The OpenEdge Command Center MCP Server uses the authentication mode configured for the OpenEdge Command Center Server:

- Local authentication with a personal access token (PAT)

- Identity provider (IdP) authentication with OpenID Connect (OIDC)

An OpenEdge Command Center Server installation uses one authentication mode at a time. Users do not select the mode in the MCP client.

Credential handling and role-based access

OpenEdge Command Center Server authenticates each MCP connection and authorizes each request. The MCP Server does not grant additional permissions or provide a separate role-based authorization system.

- In local authentication mode, access is based on the roles and permissions of the PAT owner.
- In OIDC authentication mode, access is based on the roles provided by the identity provider and enforced by OpenEdge Command Center.

The MCP Server returns only information that the authenticated user is authorized to access. It does not provide tools to modify OpenEdge Command Center resources or manage PATs, passwords, or identity-provider credentials. Manage PATs through the OpenEdge Command Center APIs.

Configure and manage the OpenEdge Command Center MCP Server

The OpenEdge Command Center MCP Server runs separately from the OpenEdge Command Center Server. It accepts MCP client connections over HTTPS on port 8002 by default and connects to the OpenEdge Command Center Server User API on port 8000 by default.

The MCP Server uses `mcpConfig.json` to define its HTTPS listener, its connection to the OpenEdge Command Center Server, and its certificate settings. The MCP client must be able to reach the MCP Server, and the MCP Server must be able to reach the OpenEdge Command Center Server User API.

The following table lists the configuration settings by section:

- `oecc_mcp` defines the MCP Server's HTTPS listener and incoming client connections.
- `oecc` defines the connection to the OpenEdge Command Center Server.
- The top-level `data` setting identifies the shared OpenEdge Command Center data directory.

Security field	Description
oecc_mcp—MCP Server HTTPS listener	
host	Specifies the network interface on which the MCP Server listens. The IP address <code>0.0.0.0</code> binds the server to all interfaces.
port	Specifies the TCP port for incoming client connections. The default is 8002.
key	Specifies the path to the private key file securing the MCP Server's HTTPS listener.
certificate	Specifies the path to the TLS certificate presented to connecting clients.

Security field	Description
intermediateCertificates	Specifies the paths to intermediate CA certificates required to complete the certificate chain, if applicable.
rootCertificates	Specifies the paths to trusted root CA certificates used to validate the certificate chain.
keyPassPhrase	Specifies the encrypted passphrase that protects the private key file. The passphrase is encrypted automatically on first startup.
allowedHosts	Specifies the host names or IP addresses permitted to access the MCP Server.
oecc—Upstream OpenEdge Command Center Server connection settings	
host	Specifies the host name or IP address of the OpenEdge Command Center Server.
port	Specifies the port on which the OpenEdge Command Center Server listens. The default is 8000.
rootCertificates	Specifies the paths to trusted root CA certificates used to validate the OpenEdge Command Center Server TLS certificate.
intermediateCertificates	Specifies the paths to intermediate CA certificates required to complete the OpenEdge Command Center Server certificate chain, if applicable.
nohostverify	Specifies whether to disable host-name verification for the OpenEdge Command Center Server TLS certificate. Set this value to <code>false</code> in production.
requestTimeoutMs	Specifies the maximum time, in milliseconds, to wait for a response from the OpenEdge Command Center Server before timing out. The default is 30000 milliseconds (30 seconds).
maxResponseBytes	Specifies the maximum response size, in bytes, accepted from the OpenEdge Command Center Server. The default is approximately 5 MB.
Top-level setting	
data	Specifies the path to the shared OpenEdge Command Center data directory, relative to the configuration file location.

Configure certificates for local and remote access

The OpenEdge Command Center Server installation includes a default set of certificates for communication among OpenEdge Command Center, the MCP Server, Agents, and clients. The default set uses a self-signed root CA, `oeccrootca.crt`; the MCP Server certificate is signed by this root CA. The default certificate configuration supports local MCP access from `localhost`.

For production deployments, Progress recommends replacing the certificates supplied with the installation with certificates issued by a certificate authority trusted by your organization. Keep certificate-chain and host-name verification enabled.

Local access with the default certificates

If you keep the default certificate configuration and run the MCP client on the same machine as the MCP Server, install the supplied root CA before starting the MCP Server or connecting from VS Code:

1. Locate the supplied root CA certificate installer.
2. Double-click the installer.
3. Select the certificate installation option and complete the installation.
4. Start the MCP Server.
5. In VS Code, connect using the local endpoint configured in `mcp.json`, with `localhost` as the host name.

Installing the root CA allows the client to verify the certificate presented by the MCP Server.

Remote access with new certificates

The default certificates are intended for local access and are not suitable for connections from a remote machine. For remote access, generate or obtain a complete new certificate set, including a root CA, server certificates, and MCP Agent certificates. Replace the applicable certificates on the relevant components.

Configure the MCP Server to use the replacement certificate and private key, and update the applicable paths in `mcpConfig.json`. Add the MCP Server host name or IP address that clients use to connect to the `allowedHosts` setting under `oecc_mcp`. The value must match the host used in the client endpoint. For example, if the endpoint uses `https://192.0.2.10:8002/mcp`, add `192.0.2.10` to `allowedHosts`.

Install or trust the new root CA on the machine running VS Code. Ensure the MCP Server certificate is valid for the host name or IP address in the client endpoint, and keep certificate-chain and host-name verification enabled. The `allowedHosts` setting permits requests for the specified host; it does not replace TLS certificate validation.

Restart the MCP Server. In VS Code, update the endpoint in `mcp.json` to use the remote host name or IP address and configured port, then connect.

If a certificate error prevents the connection, see [Troubleshoot OpenEdge Command Center MCP Server connections](#).

Start, stop, and check the OpenEdge Command Center MCP Server

The OpenEdge Command Center MCP server is installed and started automatically with the OpenEdge Command Center server. It runs as an independent operating-system process and accepts connections from MCP clients. Starting or stopping the MCP Server does not start or stop the OpenEdge Command Center Server.

Start the MCP Server

When you start the OpenEdge Command Center server, OpenEdge Command Center MCP server also starts automatically. If it is not running, or if you stopped it, you can start it by running:

- `oeccmcpserver.bat start` (on Windows)
- `oeccmcpserver start` (on Linux)

The MCP Server listens on the port specified by `port` under the `oecc_mcp` section in `mcpConfig.json`. The default port is 8002.

Stop the MCP Server

Similarly, to stop a running MCP server, you can run the following commands:

- `oeccmcpserver.bat stop` (on Windows)
- `oeccmcpserver stop` (on Linux)

Stopping the MCP server prevents MCP clients from connecting to it. It does not stop the OpenEdge Command Center server.

Check the MCP Server status

To check the status of the MCP server, you can run the following commands:

- `oeccmcpserver.bat status` (on Windows)
- `oeccmcpserver status` (on Linux)

Use the status command after starting or stopping the MCP Server to confirm whether the process is running.

Review MCP Server logs

The MCP server writes its log files to the same logs directory as the other OpenEdge Command Center Server logs. Review the logs if the MCP server does not start, stops unexpectedly, or does not accept client connections.

Verify the OpenEdge Command Center MCP Server

The OpenEdge Command Center MCP Server is included with the OpenEdge Command Center Server installer and is installed as part of the server installation. No separate MCP Server installation is required.

To verify the MCP Server:

- Check the service status:
 - On Windows, run `oeccmcpserver.bat status`.
 - On Linux, run `oeccmcpserver status`.
- Confirm that the port setting under the `oecc_mcp` section of `mcpConfig.json` is 8002.
- Confirm that the MCP client uses the following endpoint:
`http://<mcp-server-host>:8002/mcp`
- Check the MCP Server logs in the same logs directory used by the other OpenEdge Command Center Server logs. Review the logs for startup messages or errors.
- If the MCP Server is not running, start it or use the `ProgressOpenEdgeCommandCenterMcpServer2.1` service.
 - On Windows, run `oeccmcpserver.bat start`

- On Linux, run `oeccmcpserver start`

Connect to the OpenEdge Command Center MCP Server

The OpenEdge Command Center Server determines which authentication mode the MCP client must use. Before connecting, confirm whether your installation uses local authentication or OpenID Connect (OIDC) authentication.

The MCP client connects to the OpenEdge Command Center MCP Server at:

`https://<mcp-server-host>:8002/mcp`

Replace `<mcp-server-host>` with the host name or IP address of the computer where the MCP Server runs.

Follow the procedure for your authentication mode:

- **Local authentication:** See Create a PAT and connect an MCP client with it.
- **OIDC authentication:** See Connect an MCP client with OpenID Connect (OIDC).

Each procedure explains the required client configuration, how to connect, and how to verify access with a read-only request.

Create a PAT and connect an MCP client with it

When OpenEdge Command Center uses local authentication, create and manage personal access tokens through the OpenEdge Command Center APIs. PAT management is not performed through the MCP Server. You can use Swagger to call the APIs.

Create a PAT

To create a PAT, you need to send a POST request to `/api/pat` endpoint by providing the following request fields:

- `name`: Required. A name that identifies the PAT.
- `description`: Optional. A description of how the PAT will be used.
- `expiresInDays`: Specify the PAT lifetime in days, as supported by the API.

When the PAT is created, the API returns the PAT. Copy it immediately and store it securely. The PAT is not displayed again after the creation response. Do not store the PAT in documentation, source control, screenshots, or log files.

A PAT uses the roles and permissions of the user who created it. The MCP client can therefore access only the OpenEdge Command Center information that the PAT owner is authorized to view.

Revoke a PAT

To revoke a PAT, you can send a DELETE request to `/api/pat/<patId>` endpoint. Replace `<patId>` with the identifier of the PAT to revoke. Revocation prevents the PAT from being used for a new MCP connection.

Respond to an exposed PAT

Treat an exposed PAT as compromised. Do not continue using it.

1. Revoke the PAT through the OpenEdge Command Center API:

```
DELETE /api/pat/<patId>
```

Replace <patId> with the identifier of the exposed PAT.

2. Remove the exposed PAT from the MCP client configuration and any other location where it was stored.
3. Review where the PAT was exposed and remove it from those locations where possible.
4. Create a replacement PAT through the OpenEdge Command Center API:

```
POST /api/pat
```

5. Copy the new PAT from the creation response and store it securely.
6. Update the MCP client configuration with the new PAT.
7. Reconnect the MCP client and run a read-only request to verify the connection.

PAT creation, revocation, and authentication are performed through OpenEdge Command Center APIs. The MCP Server does not provide tools for creating, listing, updating, or revoking PATs.

Connect an MCP client with a PAT

Use the following configuration when OpenEdge Command Center uses local authentication. The MCP Server listens on port 8002 by default. If the port was changed, use the value specified by `port` under the `oecc_mcp` section in `mcpConfig.json`.

```
{
  "servers": {
    "oecc_mcp": {
      "type": "http",
      "url": "https://<mcp-server-host>:<mcp-server-port>/mcp",
      "headers": {
        "Authorization": "Bearer <Personal Access Token>"
      }
    }
  }
}
```

Replace <mcp-server-host> with the MCP Server host name or IP address, <mcp-server-port> with the configured MCP Server port, and <Personal Access Token> with the PAT created through the OpenEdge Command Center APIs.

Here, the server name `oecc_mcp` is an example. You can use another name for the server entry. Save the `mcp.json` file. Reload or refresh the MCP configuration if required by the MCP client.

Connect and verify the client

1. Start or connect to the server entry configured in `mcp.json`.
2. Wait for the MCP client to report that the MCP Server is connected.
3. Run the following read-only request:

```
List the OpenEdge Command Center databases that I am authorized to view.
```

The MCP Server returns only the databases that the PAT owner is authorized to view.

Connect an MCP client with OpenID Connect (OIDC)

Use this procedure when OpenEdge Command Center uses OIDC authentication.

Prerequisites

Before configuring the MCP client, ensure that:

- The OpenEdge Command Center Server is configured to use OIDC authentication.
- An identity provider is configured for OpenEdge Command Center.
- A client ID is configured for OpenEdge Command Center.
- The MCP Server is running and available over HTTPS.
- The MCP client can trust the MCP Server certificate.

For information about configuring the identity provider, see [Switch from local authentication to IdP authentication](#).

Configure the MCP client

Open the MCP client configuration file, such as `mcp.json`, and add the following server configuration. The MCP Server listens on port 8002 by default. If the port was changed, use the value specified by `port` under the `oecc_mcp` section in `mcpConfig.json`.

```
{
  "servers": {
    "oecc_mcp": {
      "type": "http",
      "url": "https://<mcp-server-host>:<mcp-server-port>/mcp",
      "oauth": {
        "clientId": "<Client ID>"
      }
    }
  }
}
```

Replace `<mcp-server-host>` with the MCP Server host name or IP address, `<mcp-server-port>` with the configured port, and `<Client ID>` with the client ID configured for OpenEdge Command Center.

Save the `mcp.json` file.

Do not add a PAT to this configuration when OIDC authentication is active.

Connect and verify the client

1. Start or connect to the server entry configured in `mcp.json`.
2. Complete the identity-provider authentication required by the MCP client.
3. Wait for the MCP client to report that the MCP Server is connected.
4. Run the following read-only request:

```
List the OpenEdge Command Center databases that I am authorized to view.
```

OpenEdge Command Center determines which databases the authenticated user is authorized to view. The MCP Server does not provide additional permissions.

Troubleshoot OpenEdge Command Center MCP Server connections

Use these checks when an MCP client cannot connect to the OpenEdge Command Center MCP Server or a tool request fails.

Issue	Resolution
The MCP Server does not start	• Check the MCP Server status. On Windows, run <code>oeccmcpserver.bat status</code>. On Linux, run <code>oeccmcpserver status</code>. • Review the MCP Server messages in the same logs directory used by the other OpenEdge Command Center Server logs. Check for invalid configuration settings, missing certificate files, private-key errors, or other startup errors. • If the MCP Server is not running, start it with <code>oeccmcpserver.bat start</code> on Windows or <code>oeccmcpserver start</code> on Linux. Check its status again.
The MCP client cannot reach the MCP Server	• Confirm that the client can reach the MCP Server host and that the port specified by <code>oecc_mcp.port</code> is available. The default port is 8002. • Use the configured host and port with the <code>/mcp</code> path in the client endpoint: <code>https://<mcp-server-host>:<configured-port>/mcp</code> • Confirm that the host name or IP address used in the MCP client endpoint is included in <code>allowedHosts</code> under <code>oecc_mcp</code> in <code>mcpConfig.json</code>. • If you change <code>oecc_mcp.port</code> or <code>allowedHosts</code>, restart the MCP Server. If you change the port, update the endpoint in the MCP client configuration. Check that network and firewall rules allow the connection.

Issue	Resolution
The MCP client cannot establish a TLS connection	• Confirm that the MCP client trusts the certificate presented by the MCP Server. Check the certificate and private-key paths under <code>oecc_mcp</code> in the MCP Server configuration. If the certificate or key has been replaced, confirm that the configuration points to the replacement files and restart the MCP Server. • For remote connections, confirm that the certificate is valid for the host name used in the client endpoint. Keep certificate verification enabled.
The OpenEdge Command Center Server is unreachable	• Confirm that the OpenEdge Command Center Server is running and that the MCP Server host can reach it. Check the <code>oecc.host</code> and <code>oecc.port</code> settings in the MCP Server configuration. The default OpenEdge Command Center Server port is 8000. • If the logs indicate an upstream TLS failure, check the root and intermediate certificates configured under <code>oecc</code>. The <code>oecc.nohostverify</code> setting applies to the MCP Server's connection to the OpenEdge Command Center Server; it does not control whether an MCP client trusts the MCP Server certificate. Keep host-name verification enabled in production.
Authentication fails	Use the client configuration that matches the authentication mode active on the OpenEdge Command Center Server: • Local authentication: Confirm that the personal access token (PAT) in the client configuration is valid and has not expired or been revoked. • OpenID Connect (OIDC) authentication: Confirm that the client ID is configured correctly and that the required identity-provider authentication completes. Do not use a PAT when OIDC authentication is active.

Issue	Resolution
Authorization fails	OpenEdge Command Center Server determines what the authenticated user can access. Confirm that the user has permission to view the information requested by the tool. In local mode, check the PAT owner's roles and permissions. In OIDC mode, check the roles provided through the configured identity provider.
A tool request fails	Review the error shown by the MCP client and the corresponding MCP Server log messages. Depending on the error, check the tool inputs, server availability, certificate trust, active authentication mode, and user permissions. Correct the identified issue and run the request again.

OpenEdge Command Center utilities

This chapter describes the OpenEdge Command Center utilities in alphabetical order. It describes the purpose, syntax, and parameters for each utility.

For details, see the following topics:

- [OECCAGENT utility](#)
- [OECCSERVER utility](#)
- [RESETOECCDBPASSWORD utility](#)
- [RESETSUPERADMIN utility](#)
- [SWITCHAUTHMODE utility](#)

OECCAGENT utility

OpenEdge Command Center provides the `oeccagent` utility to manage the OpenEdge Command Center agent. You can use this utility to start or stop the agent as a process for planned maintenance, troubleshooting, and so on.

This utility is located in the directory where the OpenEdge Command Center agent is installed.

Syntax

```
oeccagent [start | stop]
```

Parameters

start

Starts the OpenEdge Command Center agent as a process.

stop

Stops the OpenEdge Command Center agent.

OECCSERVER utility

OpenEdge Command Center provides the `oeccserver` utility to manage the OpenEdge Command Center server and MongoDB. You can use this utility to start or stop these components individually or together for planned maintenance, troubleshooting, and so on.

This utility is located in the directory where the OpenEdge Command Center server is installed.

Syntax

```
oeccserver [start | stop | startdatabase | stopdatabase]
```

Parameters

start

Starts the OpenEdge Command Center server as a process and MongoDB.

stop

Stops only the OpenEdge Command Center server.

startdatabase

Starts only MongoDB.

stopdatabase

Stops only MongoDB.

RESETOECCDBPASSWORD utility

OpenEdge Command Center provides the `resetoeccdbpassword` utility, which allows administrators to reset the password for the OpenEdge Command Center database user.

Before resetting the password, ensure that the following prerequisites are met:

- You must run this utility only on the primary server in a high-availability setup or on a standalone server.
- The OpenEdge Command Center database is running on the primary server in a high-availability setup or on a standalone server.

The `resetoeccdbpassword` utility is located in the `utils` directory within the directory where the OpenEdge Command Center server is installed.

For example,

- On Windows, `C:\Progress\OECC\Server\utils`
- On Linux, `/Progress/OECC/server/utils`

Syntax

```
resetoeccdbpassword
```

When you run the utility, it interactively prompts for a new password and confirmation. The utility masks the password as you enter it.

Note: The new password cannot be empty.

For example:

```
resetoeccdbpassword.bat
Starting OECC DB Password Reset Utility...
Enter new password:****
Confirm new password:****
OECC DB user password updated successfully

Success: OECC DB password has been reset.
```

After the password is updated:

- The utility automatically updates the encrypted database password stored in the `db-config.json` file.
- The server automatically reconnects to the database. In a high-availability setup, all server instances automatically reconnect to the database.

RESETSUPERADMIN utility

OpenEdge Command Center provides the `resetsuperadmin` utility to manage the superuser administrator details. As an administrator, you can use this utility to modify the following details:

- First name
- Last name
- Password
- Email address
- Description

The `resetsuperadmin` utility is located in the `utils` directory within the directory where the OpenEdge Command Center server is installed.

For example,

- On Windows, C:\Progress\OECC\Server\utils
- On Linux, /Progress/OECC/server/utils

For more information, see [Reset super admin user details](#) on page 103.

Syntax

```
resetsuperadmin [--firstname <first name>] | [--lastname <last name>] | [--pwd  
<password>] | [--email <email>] | [--description <description>]
```

For example:

```
resetsuperadmin [--firstname John] | [--lastname Smith] | [--pwd P@ssw0rd123!] |  
[--email superadmin@gmail.com] | [--description Primary user of the OpenEdge  
Command Center server]
```

Parameters

--firstname <first name>

Specifies the first name to be updated. Only alphanumeric characters are allowed.

For example, --firstname "John"

--lastname <last name>

Specifies the last name to be updated. Only alphanumeric characters are allowed.

For example, --lastname "Smith"

--pwd <password>

Specifies the new password for the user account. The password must meet the following requirements:

- length of 8 to 40 characters
- A mixture of uppercase and lowercase letters
- At least one numeral
- At least one of the special characters: ! @ \$ % ^ () _ + = [] |

For example, --pwd "P@ssw0rd123!"

--email <email>

Specifies the new email address for the user account.

For example, --email "superadmin@gmail.com"

--description <description>

Specifies the description for the user account.

For example, `--description "Primary user of the OpenEdge Command Center server"`

SWITCHAUTHMODE utility

You use the SWITCHAUTHMODE utility on the OpenEdge Command Center server host to store and activate authentication configurations, and to return to local authentication when required.

Syntax

```
switchauthmode --list
switchauthmode --add
switchauthmode --activate <configuration-name>
switchauthmode --remove <configuration-name>
switchauthmode --activate local
```

When you run SWITCHAUTHMODE `--ADD`, the utility prompts you to enter the following configuration values interactively, one at a time:

- configuration name
- provider (entra, okta, or keycloak)
- issuer URL
- client ID
- client secret
- redirect URI
- scope, including the custom API scope value
- roles claim path (optional)

Caution: Use SWITCHAUTHMODE `--REMOVE` only to delete a stored but inactive configuration.

List and remove configurations

To review the stored configurations, including the active one, run:

```
switchauthmode --list
```

To delete a stored configuration that is not active, run:

```
switchauthmode --remove "Dev Okta"
```

Note: To delete the active configuration, first activate another configuration.

Parameters

Parameter	Required	Description
--list	No	Lists the stored authentication configurations and identifies the active one.
--add	No	Stores a configuration. The configuration is inactive until you activate it.
--activate	No	Activates a named configuration. Restart the server to apply the change.
--remove	No	Deletes the named configuration. You cannot delete the active configuration.
--name	Yes, with --add	Name of the configuration, for example, Company Entra ID. Name is the key that you use to activate or remove the configuration.
--provider	Prompted if omitted	Identity provider type. The supported values are <i>entra</i> and <i>okta</i> .
--issuerUrl	Yes, with --add	OIDC issuer URL. OpenEdge Command Center reads the endpoints of the identity provider from the OpenID Connect discovery document that is published at this URL, so you do not configure those endpoints individually.
--clientId	Yes, with --add	Application or client ID from the identity provider. The value is stored unencrypted.
--rolesClaimPath	No (optional)	Specifies the claim path in the IdP access token that contains the user's assigned roles.
--clientSecret	Yes, with --add	Client secret from the identity provider. The value is stored in encrypted form.
--redirectUri	Yes, with --add	Callback URL of the OpenEdge Command Center server. The value must match the redirect URI that is registered in the identity provider exactly. The path is <code>/api/auth/callback</code> .
--apiScope	No	Custom API scope that makes the identity provider issue a JWT access token that contains a roles claim. Required for both Microsoft Entra ID and Okta.
--scope	No	Space-separated OpenID Connect scopes requested from the identity provider. The default is <code>openid profile email offline_access</code> .

Notes

- The utility must run with root privileges on Linux, or with Administrator privileges on Windows, otherwise it exits with an error and makes no change.
- The utility is available only on the OpenEdge Command Center server host. The authentication mode cannot be changed from the interface or from the REST API.
- Activating a configuration does not take effect until you restart the OpenEdge Command Center server.

- Activating an identity provider configuration sets all users to inactive, and activating the local configuration restores them and requires each user including admin to set a new password.

Roles claim path: Nested-scenario examples

The `rolesClaimPath` is an optional field. It specifies the path to the claim that contains the user's roles in the access token. It uses the dot-notation syntax, so it can reference nested objects and array elements. This value is useful when the roles are not stored in a default claim such as `roles`, `resource_access.<clientId>.roles`, or `realm_access.roles`.

Example 1: One level of nesting

Token:

```
{
  "custom_claims": {
    "app_permissions": ["role:oecc/oecc_admin"]
  }
}
```

Path:

```
custom_claims.app_permissions
```

Example 2: Per-client nesting with the {clientId} placeholder

Token:

```
{
  "app_access": {
    "oecc-server": {
      "roles": ["role:oecc/oecc_admin"]
    }
  }
}
```

Path:

```
app_access.{clientId}.roles
```

At run time, this path resolves to `app_access.oecc-server.roles` by using the client ID from the same configuration.

Example 3: Nested object with an array index

Token:

```
{
  "groups": [
    {
      "name": "primary",
      "roles": ["role:oecc/oecc_admin"]
    }
  ]
}
```

Path:

```
groups[0].roles
```

If none of these paths resolves, for example because of a typo or because the claim is not present in a given token, OpenEdge Command Center falls back to the built-in claim shapes (`roles`, `resource_access.<clientId>.roles`, and `realm_access.roles`) automatically. It does not fail or error out.

Examples

Add and activate a Microsoft Entra ID configuration

The `--add` command creates an OIDC configuration and prompts for its values interactively.

```
switchauthmode --add
```

Then provide the following values when prompted:

```
Configuration name: Company Entra ID
Provider: entra
Issuer URL: https://login.microsoftonline.com/<tenant-id>/v2.0
Client ID: <Application (client) ID>
Client secret: <Client secret value>
Redirect URI: https://<oecc-server>:<port>/api/auth/callback
Scope: api://<client-id>/<exposed-custom-scope>
Roles claim path: optional, for example app_access.{clientId}.roles
```

Then run:

```
switchauthmode --activate "Company Entra ID"
```

Add and activate an Okta configuration

1. Add and activate an Okta configuration that uses a custom authorization server:

```
switchauthmode --add
```

2. Provide the following values when prompted:

```
Configuration name: Dev Okta
Provider label: okta
Issuer URL: https://<your-okta-domain>.okta.com/oauth2/<authServerId>
Client ID: <Okta client ID>
Client secret: <input hidden – not echoed>
Redirect URI: https://<oecc-server>:<port>/api/auth/callback
Scope: <custom-scope-name> and optional OIDC scopes, such as openid profile email
offline_access
Roles claim path: optional
```

3. Run the following:

```
switchauthmode --activate "Dev Okta"
```

OpenEdge Command Center performance results

The following information shows performance results of OpenEdge Command Center under various scenarios. Note that actual performance may vary depending on workload, network conditions, system tuning, and so on.

Test system configuration

- RAM: 32 GB
- CPU: 8-core CPU
- Hard disk: 100 GB

Performance results

- Agent-level performance—The tests evaluated a single agent managing different combinations of OpenEdge resources.
 - PAS for OpenEdge instances—The agent was able to manage up to 40 PAS for OpenEdge instances.
 - OpenEdge databases—The agent was able to manage up to 800 OpenEdge databases.
 - Both PAS for OpenEdge instances and OpenEdge databases—The agent was able to manage up to 30 PAS for OpenEdge instances and 200 OpenEdge databases.
- Server-level performance—The tests evaluated a single server managing multiple agents with a varying number of OpenEdge resources.
 - The server was able to manage 10 to 15 agents.

Performance varies based on the number of resources, including PAS for OpenEdge instances, ABL applications, web applications, and OpenEdge databases that each agent handles. You can use these performance results to plan and optimize your OpenEdge Command Center deployments.