Progress®



Achieve Continuous Operation with OpenEdge

OpenEdge®

Copyright

© 2021 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.

These materials and all Progress® software products are copyrighted and all rights are reserved by Progress Software Corporation. The information in these materials is subject to change without notice, and Progress Software Corporation assumes no responsibility for any errors that may appear therein. The references in these materials to specific platforms supported are subject to change.

DataDirect Cloud, DataDirect Connect, DataDirect Connect64, DataDirect XML Converters, DataDirect XQuery, DataRPM, Defrag This, Deliver More Than Expected, DevReach (and design), Icenium, Inspec, Ipswitch, iMacros, Kendo UI, Kinvey, MessageWay, MOVEit, NativeChat, NativeScript, OpenEdge, Powered by Chef, Powered by Progress, Progress, Progress Software Developers Network, SequeLink, Sitefinity (and Design), Sitefinity, Sitefinity (and design), SpeedScript, Stylus Studio, Stylized Design (Arrow/3D Box logo), Styleized Design (C Chef logo), Stylized Design of Samurai, TeamPulse, Telerik, Telerik (and design), Test Studio, WebSpeed, WhatsConfigured, WhatsConnected, WhatsUp, and WS_FTP are registered trademarks of Progress Software Corporation or one of its affiliates or subsidiaries in the U.S. and/or other countries.

Analytics360, AppServer, BusinessEdge, Chef Automate, Chef Compliance, Chef Desktop, Chef Habitat, Chef WorkStation, Corticon.js, Corticon Rules, Data Access, DataDirect Autonomous REST Connector, DataDirect Spy, DevCraft, Fiddler, Fiddler Everywhere, FiddlerCap, FiddlerCore, FiddlerScript, Hybrid Data Pipeline, iMail, JustAssembly, JustDecompile, JustMock, KendoReact, NativeScript Sidekick, OpenAccess, PASOE, Pro2, ProDataSet, Progress Results, Progress Software, ProVision, PSE Pro, Push Jobs, SafeSpaceVR, Sitefinity Cloud, Sitefinity CMS, Sitefinity Digital Experience Cloud, Sitefinity Feather, Sitefinity Insight, Sitefinity Thunder, SmartBrowser, SmartComponent, SmartDataBrowser, SmartDataObjects, SmartDataView, SmartDialog, SmartFolder, SmartFrame, SmartObjects, SmartPanel, SmartQuery, SmartViewer, SmartWindow, Supermarket, SupportLink, Unite UX, and WebClient are trademarks or service marks of Progress Software Corporation and/or its subsidiaries or affiliates in the U.S. and other countries. Java is a registered trademark of Oracle and/or its affiliates. Any other marks contained herein may be trademarks of their respective owners.

July 2021

Product version: Progress OpenEdge 12.4

Preface

Purpose

This guide provides an overview of how to achieve continuous operation with OpenEdge. It describes the key components in OpenEdge that enable your application to operate continuously, lists the main steps involved in using ABL or SQL applications, configuring your database, configuring PAS for OpenEdge, setting up data replication, using a load balancer, and monitoring your platform for potential issues.

Audience

This guide is aimed at OpenEdge solution architects, application developers, and database administrators who want to learn about continuous operation with OpenEdge.

Organization

This guide consists of the following parts:

- Learn about continuous operation with OpenEdge on page 7: Provides an overview of continuous operation in OpenEdge.
- Operate continuously using OpenEdge database features on page 11: Describes the continuous operation features by design.
- Set up Automatic Database Reconnect on page 17: Describes the features and products that you may want to develop for continuous operation.
- Configure your database on page 21: Describes the features and products that you may want to configure for continuous operation.
- Learn about monitoring on page 27: Provides an overview of how you can use monitoring for continuous operation.

Documentation conventions

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

Get started

For details, see the following topics:

- Learn about continuous operation with OpenEdge
- Continuous operation basics

Learn about continuous operation with OpenEdge

Continuous operation with OpenEdge is the full utilization of OpenEdge software suite to achieve your application architecture goals.

A continuously operational application architecture protects enterprises from lost revenue by keeping their data and applications available, even if access to resources is disrupted. This is a necessity for businesses across the globe. As more enterprises want to extend the availability of their mission-critical applications and data, they are under pressure to ensure the up-time of their systems. Downtime, even if it is a matter of minutes, can result in substantial costs to the enterprise. To ensure the maximum feasible amount of uptime, an enterprise must have several components in their architecture: the ability to scale according to changing requirements and loads, the establishment of replicated sets of characteristics when parts of the application crash, the reduction of maintenance downtime, the increase of error detection and resolution speed, and fast recovery without data loss in the event of unexpected downtime.

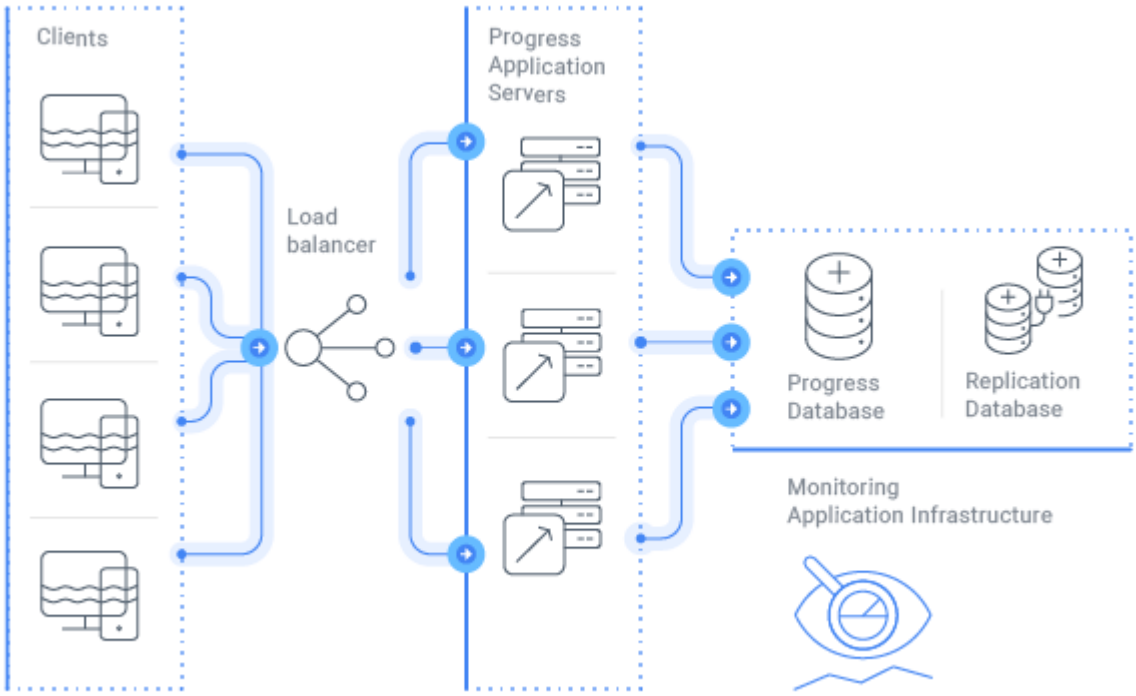
By design, OpenEdge offers a set of features, tools, and parameters, both out of the box and configurable, to achieve continuous operation such as database parameters that are modifiable online, database replication, the OpenEdge HealthScanner, and more.

Leveraging these features, OpenEdge can:

- Protect an enterprise from downtime.
- Save lost revenue.
- Simplify maintenance.
- Maximize flexibility.
- Harden resilience.

Architecture for OpenEdge continuous operation

The following illustration presents an OpenEdge configuration optimized for continuous operation and monitoring.



This OpenEdge architecture includes the following key components:

- **OpenEdge database**—OpenEdge databases are designed with continuous operation in mind, and provide ways to update parameters, schemas, and more while online.
- **Data replication**—Data replication helps you avoid single points of failure by enabling a disaster recovery or a failover site. Data replication ensures the availability of an application, and distributes copies of data to one or more sites for horizontal read scaling. Using database replication is the best way to make your application highly available and fault tolerant. If your production database, database machine, or network becomes unavailable you can quickly transition to the replicated database.
- **OpenEdge clients**—OpenEdge supports three types of clients that use HTTP networking. Each client uses HTTP networking differently. How these clients communicate with a PAS for OpenEdge instance impacts continuous operations different ways:
  - ABL, Java, .Net, and WebClients employ a point-to-point network connection to a PAS for OpenEdge instance. That point-to-point connection may be load balanced, where each new connection can be directed to any active PAS for OpenEdge instance. However, a client cannot switch to another PAS for OpenEdge instance if the original PAS for OpenEdge instance goes out of service. The client needs to initiate a new connection.
  - HTTP clients access a PAS for OpenEdge instance's REST, Web, and web applications using connectionless HTTP. Therefore, switching client message destinations between PAS for OpenEdge instances does not impact the continuous operation.
  - SOAP services use either a session-managed or session-free model. The session-managed model uses a point-to-point connection to PAS for OpenEdge instances. The client is disabled until the connection is resumed. The session-free model uses connectionless message delivery of HTTP to take full advantage of continuous operation.
- **Progress Application Server (PAS) for OpenEdge**—PAS for OpenEdge should be configured for scalability and enabled to work seamlessly with load balancers for continuous operation.
- **Load balancer**—A full-featured load balancer is a required part of the OpenEdge continuous operation solution. OpenEdge does not provide a load balancer, so finding the correct load balancer with the needed features is crucial. The load balancer redirects traffic from any PAS for OpenEdge instance that runs the risk of becoming over loaded to an instance with less traffic.
- **Monitoring tools**—To have a continuously operating application, an enterprise must constantly monitor their application's ecosystem, and have a system of alerts to warn about issues. Monitoring the application ecosystem allows an enterprise to observe and act on possible issues before they happen. With OpenEdge, you can monitor your applications with the OpenEdge HealthScanner, OpenEdge Management, and deferred logging.
- **(Optional) Docker and Kubernetes with PAS for OpenEdege**—Building your applications to run in a Docker container enables you to develop, deploy, and run scalable applications in an isolated environment. Encapsulating applications in a container allows for easy installation and repeatable execution. Kubernetes can be used to deploy, scale and manage containerized applications. Kubernetes can be used to quickly and easily deploy new versions of applications to end users without the need to bring down your system.

Continuous operation basics

At the foundation of continuous operation is the no-single-point-of-failure (SPOF) rule. That is, no single hardware or software component failure exists that can appear to the end user as an application outage.

That does not mean that the user does not see a failure or have to connect again, but that the user can quickly resume their previous work with minimal interruption. What you are looking for are architectural components that appear only once in your application's deployment. A good place to start is looking for and understanding those SPOFs in:

- File system data storage
- Networking components
- OS image
  - Database server
  - Application servers

The types of SPOF that you can incorporate into your ABL application's architecture are the OS images and the OpenEdge products that operate inside one. Following the basic SPOF rule, you should have a plan to have two, or more, instances of each OpenEdge component running and able to take over the load in case of failure.

Most OS images have failover or recovery for simple daemon processes, whether it is simple built-in service process monitoring or more container monitoring when your daemon services are running in that environment. Look at your OS image's feature and package support for the best fit for your ABL application's processes.

II

By design

For details, see the following topics:

- Operate continuously using OpenEdge database features
- Operate continuously using ABL application features
- Operate continuously using SQL application features

Operate continuously using OpenEdge database features

By default, OpenEdge databases are designed with continuous operation in mind. Whether you are upgrading from a previous version of OpenEdge, or building your application for the first time, OpenEdge databases are built to be robust.

While a database cannot be load balanced like the PAS for OpenEdge instances, you must ensure continual access to it. If your production database, database machine, or network becomes unavailable you can quickly transition to the replicated database with minimal outage.

ABL sessions and databases have continuous operation features that reduce the time needed to restart a database or reconnect to a database replica. When you run multiple copies of a replicated target database, if the source goes down, a target transitions to the source database.

If you have replication plus, an opt-in OpenEdge feature, you can also use a target database as a READ-ONLY database, taking that work off of the source database machine.

Another critical aspect of database continuous operation is maintaining an ABL session's database connection. To maintain that connection, use the following ABL continuous operation capabilities:

- ABL session database connection options— These options permit an automated reconnection process to initiate if the initial connection is interrupted. While this option does not permit transparent reconnections, it moves the responsibility for detecting and restoring a disconnect from your application to the OpenEdge database.
- Enhancements to the OpenEdge database that reduce the time a database is available for a client reconnection.

However, there are times when you need to perform maintenance or to push updates to your updates. In the past this meant that you might need to take your database down, causing disruption to your users. To minimize down time and, by extension, to prevent disruption to users, OpenEdge is designed to be able to push updates, schema changes, and parameter modifications to your database all while your application is still up and running.

Change database parameters while the database is online

OpenEdge offers over 100 different parameters that you can modify while your database is running, giving you versatility in an ever-changing business environment. For example, it might become necessary for you to change the maximum number of clients per server to support increased load without bringing your database down. With an OpenEdge database, you can do that and more.

For more information on modifying database parameters, see the following topics:

- Change database parameters while the database is running
- Modify database startup parameters online using PROMON

Manage indexes online

As database blocks can become fragmented, index blocks can become underused over time. The optimal degree of index block utilization depends on the type of database access performed. Retrieval-intensive applications generally perform better when the index blocks are close to full because the database engine must access fewer blocks to retrieve a record. The larger the index, the greater the potential for improving uptime by compacting the index. Update-intensive applications, on the other hand, perform better with loosely packed indexes because there is room to insert new keys without allocating more blocks. Index analysis provides the utilization information that you need to make decisions. Choose a balance between tightly packed indexes and underused indexes, depending on your data and applications.

For more information on managing indexes, see the following topics:

- 'Analyzing index use' in *Manage the OpenEdge Database*
- 'Compacting indexes' in *Manage the OpenEdge Database*
- 'Rebuilding indexes' in *Manage the OpenEdge Database*
- 'Activate a single index' in *Manage the OpenEdge Database*

Manage database brokers

To support continuous operation, OpenEdge databases log secondary broker parameters, keep secondary brokers running during recovery transitions, and let you disconnect secondary brokers without a process kill signal.

Secondary brokers let you manage the configuration for a specific group of end users such as ABL clients, SQL clients, or both.

OpenEdge features like these help you manage brokers to support continuous operation:

- **Brokers keep running during recovery transitions**—To finish transitions faster, database brokers continue running on single targets and replication sets. This improves both manual and automatic recovery transitions. For more information, see 'Plan for transition' in *Use Database Replication*.
- **You can disconnect secondary brokers using PROMON**—A PROMON option lets you disconnect secondary brokers without issuing a process kill signal.

  This option reduces potential database downtime. For more information, see 'Server Options' in *Manage the OpenEdge Database*.
- **The database log file (`.lg`) logs secondary broker parameters**—The database log file captures information that helps you understand the circumstances surrounding a failure. The log now records start up parameters for secondary brokers.

  Although you can list broker parameters for running servers by using PROMON R&D Status or Virtual System Tables (VST), the database log file lets you troubleshoot environment or connection issues after server failure. You can easily determine which brokers were active during an event.

  For an example, see 'Find connected brokers and servers' in *Manage the OpenEdge Database*.

Operate continuously using ABL application features

As you maintain your application architecture for continuous operation, you can use ABL features to make changes to schemas, triggers, fields, and more while your application and database are online to push updates without the need to bring them down.

PAS for OpenEdge application deployment was designed for supporting modern day web applications in a scalable and continuously operating environment. PAS for OpenEdge supports multiple strategies to meet various product deployment requirements. The deployment strategy that fits scalability and continuous operations best is the instance. PAS for OpenEdge instances are self-contained, easy to deploy, and scalable HTTP/S web servers that may appear in on-premise virtual machine (VM), cloud images, or Linux container deployments.

Much of PAS for OpenEdge application development's ability to support continuous operations depends on eliminating its instances of single points of failure, and enabling dynamic removal and addition of instances without client outage. In most cases HTTP load balancer products can be used to manage client traffic when PAS for OpenEdge instances go out of service, or are returned to service.

Configure schema changes online

To minimize downtime and to avoid any negative impacts on your applications, you can perform schema maintenance tasks while your database is running by using the `SCHEMA-CHANGE` attribute during an ABL session. The `SCHEMA-CHANGE` attribute relaxes the requirement for exclusive access to a database so that you can:

- Add new sequences.
- Add new tables, and any associated fields, indexes, and database triggers.
- Add new fields to an existing table.
- Add new inactive indexes to an existing table.

For more information, see Change schemas while the database is running.

Change non-structural fields while online

The non-structural fields of the `_File` and `_Field` tables have been designed to be changed while the database remains running. These field changes do not require an exclusive schema lock, enabling you to avoid disruptions in user experience. For more information, see Change non-structural fields of _File and _Field online.

Modify ABL triggers while online

In OpenEdge, you can modify the schema for your ABL trigger definitions while online and without it being necessary to acquire an exclusive lock on the entire schema. You can perform online operations to create new trigger definitions, to update, and to delete existing trigger definitions all while your users enjoy continuous access to your application. For more information, see Change ABL triggers online.

Rename fields using ABL while online

To support continuous operation, you can rename user table fields while the database is running. Renaming a database field no longer requires that you have an exclusive schema lock or that you must disconnect users from the database. This enables you to keep your users engaged with your services longer. For more information, see Rename fields using ABL.

Change Int to Int64 while online

Similar to being able to rename fields while online, you can change the datatype of a field in an application table from `Int` to `Int64` while the database is running. This change does not require exclusive access to the database.

The change takes effect without:

- Dumping and loading the affected table.
- Rebuilding indexes, if the affected field is an index key component.
- Recompiling r-code, if the data stored in the affected fields contains values in the `Int` range.

For more information, see Change Int to Int64 online.

Operate continuously using SQL application features

By design, OpenEdge offers essential flexibility in developing software solutions by providing a high-performance relational database and an open, standards-based interface for SQL.

The OpenEdge database and its SQL interface provide an environment that enables efficient integration with third-party tools, such as development software, reporting, and Online Analytical Processing (OLAP) tools.

For continuous operation of your SQL application and OpenEdge database, OpenEdge provides ways to rename fields, delete triggers, and modify schemas while your OpenEdge application is online and fully operational. You can use these features to minimize downtime for maintenance and perform updates that would normally require your application to be down.

Change field names while online

User table fields can be renamed while the database is in use. Renaming a database field, by design, does not require exclusive access to the database or for users to be disconnected from the database. When a user table field is renamed, all connected clients are notified of the change.

This functionality is specific to SQL clients in this release of OpenEdge. When a field is renamed, the SQL client that executes the change checks whether any clients of an earlier version are connected to the database. If an earlier version is found, then the process is treated as an offline operation. For more information, see Rename field using SQL.

Delete while triggers online

As a part of good application maintenance, you should delete triggers that become problematic or are no longer needed. Regular maintenance can help streamline your application to prevent unplanned downtime. By design, with OpenEdge you can delete triggers in SQL applications without bringing down your application. For more information about how to delete triggers, see Drop a trigger.

Update SQL applications while connected to the database

In the effort to limit downtime, whether it is planned or unplanned, OpenEdge provides you with the ability to use SQL commands to modify schemas while the database is online. There are a wide variety of commands that you can issue, to ensure that every business need is met. You can alter tables, create tables and indexes, delete items, set encryption, and more. Fore more information, see SQL commands to modify schema.

III

Develop

For details, see the following topics:

- Set up Automatic Database Reconnect
- Run PAS for OpenEdge in containers

Set up Automatic Database Reconnect

While there are many features for continuous operation included in ABL applications by design, it is important to develop a strategy for a potential application to database connection loss.

OpenEdge provides a feature that automatically connects your ABL application to an alternative database if the connection to the primary database fails. The Automatic Database Reconnect feature uses database connection parameters to specify alternative databases to which you application can connect. The collection of primary and alternative databases, along with an optional retry count and pause interval, is known as a *connection set*. An alternative database is typically a replication target database, but it could also be a backup or a hot standby database. This feature is only available when using a client-server connection. It is not available for a single-user or a self-service (shared-memory) connection. For more information, see Learn about automatic database reconnect.

To set up automatic database reconnection, you define a connection set consisting of the primary database and up to two alternative databases. These connection options are implemented using session startup parameters `-dbalt1`, `-dbalt2`, `-retryConnect`, and `-retryConnectPause`, in the `CONNECT` statement, or in the command-line startup parameters.

In the following video, you can view a demo of how to use the automatic database reconnect for ABL clients.

Automatic Database Reconnect for ABL Clients

For more information, see Automatic database reconnect demo and code.

Run PAS for OpenEdge in containers

Using a container architecture for an OpenEdge application allows you to upgrade individual components of the application without disrupting the application's availability.

Traditionally, an on-premises OpenEdge environment was built using multiple machines running multiple instances of PAS for OpenEdge and OpenEdge applications. Upgrading this environment meant taking down an entire production application leading to a disruption of service. In addition, the actual upgrade was a time consuming, manual process.

Docker containers can now be used to build in the redundancy and resiliency you need in an OpenEdge environment. By refactoring your OpenEdge environment into containers, a single virtual machine can be used to run multiple containers, and by extension, multiple instances of PAS for OpenEdge and your OpenEdge application. If it becomes necessary to upgrade container, you simply shut down and remove that container, and bring up a new one online with little to no impact to the operation of your production OpenEdge application.

Container architecture for on-premises and hybrid-cloud based solutions

To create an on-premises architecture for continuous operations with OpenEdge and containers, you create PAS for OpenEdge instances in containers. Each container can be shut down and removed in isolation, meaning that by making changes to the containers you do not affect the other instances or the OpenEdge application.

The components of a container-based on-premises architecture are:

- PAS for OpenEdge HTTP Clients
- PAS for OpenEdge instances running in containers
- A local load balancer
- A locally installed version of the OpenEdge database for deployment and continuous integration
- OpenEdge database replication

To manage the traffic to the PAS for OpenEdge instances, select a software load balancer that is optimized to function with multiple containers. Moreover, your load balancer you should be one that is best suited for your container execution platform. The on-premise architecture provides continuous operation through the simplification of the upgrade process.

A hybrid-cloud architecture combines both the on-premises and cloud architectures, and gives you the added advantage of continuous integration. In this architecture, the OpenEdge database and the load balancer are installed on-premises but the PAS for OpenEdge instances and web clients run in Docker containers deployed to a cloud platform, such as Amazon Web Services. When your OpenEdge application is ready for production, you integrate the containers into an automation pipeline. You now can easily upgrade both the PAS for OpenEdge instances and the clients while preserving continuous integration. This type of architecture is best suited for SaaS offerings.

For more information about automating your OpenEdge application, see Learn about CI/CD.

Container architecture for cloud-based solutions

In a cloud environment, all aspects of your deployment environment, such as your PAS for OpenEdge instances, ABL business application, and web applications, are built in containers and deployed to an automation pipeline. You utilize the native services of the cloud such as orchestration and load balancing to provide continuous operation of your OpenEdge application. As with the on-premises and hybrid-cloud architecture, each container can be modified independently of the others, and thereby do not affect the other instances or the availability for your application.

The components of an OpenEdge container architecture in the cloud are:

- Availability zones or regions built into your cloud deployment for redundancy and high availability
- PAS for OpenEdge HTTP Clients running in containers
- PAS for OpenEdge instances running in containers
- The orchestration service offered by the cloud (for example, Kubernetes)
- A load balancer service offered by the cloud platform (such as the Elastic Load Balancer)
- Locally installed OpenEdge database
- OpenEdge database replication

Refactoring your OpenEdge application into containers has the added benefit of a built-in high availability infrastructure and external availability management. As a business that uses cloud services, you do not need to worry about securing and maintaining the infrastructure for continuous operation, one is provided to you by the cloud service of your choosing.

For more information about using containers and other cloud features in OpenEdge applications, see Learn about modernization in OpenEdge.

For more information about automating your OpenEdge application, see Learn about CI/CD.

IV

Configure

For details, see the following topics:

- Configure your database
- Avoid downtime with PAS for OpenEdge
- Use a load balancer with OpenEdge

Configure your database

To configure your database for continuous operation, you need to use database network connections, replicate your database, use replication sets, and ensure the ABL application that connects to the OpenEdge database use automatic database reconnection.

Database network connections

Any ABL application with the goal of continuous operations must use OpenEdge database network connections. Network connections permit a database to continue running on a separate set of shared OS images while PAS for OpenEdge instances and ABL sessions are added and removed from service. Using network connections also protects a database in the event any ABL session client disconnects while still holding a latch or resource in the database.

Part of configuring your database is ensuring it has enough user connections to service all of its ABL session clients. This configuration must be large enough to handle the maximum number of ABL Sessions for each instance in the load balancer pool. If you choose to have a hot-standby PAS for OpenEdge instance, your database configuration must also account for those additional connections. This strategy allows for quickly scaling up PAS for OpenEdge instances without having to create a larger database to handle the extra processing.

In most deployments, the OpenEdge database is running a different OS image than the PAS for OpenEdge instances. Best practice is to remove other applications and processes from the database machine, so they don't adversely affect the performance of the database if a sudden spike in usage occurs.

Replicate your database

Using database replication is the best way to make your application continuously available and fault tolerant. If for any reason your production database, database machine, or network becomes unavailable you can quickly transition to the replicated database.

Data replication has two main real-time functions: to provide a disaster recovery or a failover site to ensure availability of an application and to distribute copies of data to one or more sites for horizontal read scaling. If you run multiple copies of replicated target databases, and if the source goes down, a target will transition to source and there will still be a target database. OpenEdge Replication provides near, real-time replication of an OpenEdge database from one location to another. The second location can be local to the production database, at a remote data center, or in the cloud.

OpenEdge Replication requires a minimum of two databases: a source database and a target database. OpenEdge can also replicate to a tertiary database. The source database is where the users do their work and make database updates. This database is the production database and has both read and write access for all processes.

The following are the criteria for an OpenEdge Replication source database:

- The source database must be enabled as an OpenEdge Replication source database.
- OpenEdge Replication server replicates data to the target database from the enabled source database.
- After-imaging is required to be enabled on the database.
- After-imaging is used to capture all database activity.

The target database originates from the source database, and contains the same data and schema as the source. The target database does not allow database updates by anything other than OpenEdge Replication. OpenEdge Replication Plus permits queries and reports by users, and any other non-database write activity. The OpenEdge Replication agent updates this database. For the most fault-tolerant deployment, you set two target databases.

The following are criteria for an OpenEdge Replication target database:

- The target database must be enabled as an OpenEdge Replication target database.
- OpenEdge Replication agent replicates data from the source database.
- The OpenEdge Replication agent is the only process that can update the database.

Replication sets

A Replication Set is a two-target Replication configuration where the two targets are aware of each other, and are configured to continue replication in the event of a server failure.

Adding a Replication Set, to an OpenEdge Replication configuration, changes the failover activity. The biggest benefit of a Replication Set is the ability to transition all available databases together. The goal of the Replication Set is to minimize the risk of a single point of failure and the need to rebase.

With a Replication Set, if a Replication agent loses communication contact with the Replication server (for example, if the primary machine were to shut down unexpectedly), you can move all database update activity from the source database one of the target databases, and the other target becomes an agent to the new source.

For more information about OpenEdge Replication, see Introduction to OpenEdge Replication.

Avoid downtime with PAS for OpenEdge

Along with expectations for the database to be updated online, users do not want to experience downtime for a new version of an application to be deployed. PAS for OpenEdge provides fault-tolerant and persistent ABL and SQL application sessions. These enhancements provide online capabilities to enable updates and troubleshooting without forcing users to disconnect from the instance.

Deploy your application while online

OpenEdge system administrators can securely deploy new ABL applications to production PAS for OpenEdge instances without having users experience any downtime. Secure ABL application deployment to a production instance requires a securely deployed Tomcat Manager, and other security considerations.

For example, as an e-commerce enterprise you want to set up your ABL applications to support microservices using PAS for OpenEdege. You need to update your applications frequently to keep your services competitive; however, your previous application architecture required a server restart when installing new microservices. Restarting your instance disrupted your other services because they rely on one another. To combat this disruption, you can choose to use PAS for OpenEdge to help build your architecture and to be able to install new versions of microservices without requiring a restart.

Main steps: Deploy applications while online

1. Create a production PAS for OpenEdge instance.
2. Deploy the Tomcat Manager web application.
3. Enable the remote IP address valve.
4. Deploy your initial web application.
5. Deploy your service to an online instance of PAS for OpenEdge..

For more information about deploying your application while online, see Secure online deployment of a new ABL application.

Update PROPATH while online

Similar to using the PAS for OpenEdge capability to deploy a new application while online, you can dynamically update applications by enabling live PROPATH updates in production instances without restarting the server.

This feature allows system administrators to:

- Deploy new application r-code to a running server instance.
- Change the PROPATH by using the OEPROP utility or by deploying an updated application to include the new r-code location.

Consider the following scenario. As an enterprise, you have an application running on a deployed instance of PAS for OpenEdge and need to push an update to that application. The application is configured as modules using procedure libraries with a versioned directory structure. To push that update using PROPATH, you can deploy a new procedure library and update the instance PROPATH, making the update available while your application is running. However, after you push the update, you realize that there is now a bug in your application. To fix this bug, you can deploy an r-code file as a hotfix to address the bug in the update you just deployed, without ever restarting the server.

For more information, see Update PROPATH in a production instance with zero downtime.

Refresh agents in an ABL application

To prevent your application architecture going down from memory growth due to application leaks, you can refresh agents in ABL applications on a PAS for OpenEdge server. Using the refresh agents API you can return memory to the operating system and free memory currently used in the agents so that it can be reused by new ABL sessions. You might notice that the refresh agents API provides operations similar to the PAS for OpenEdge's ability to update PROPATH while online, however, the API does not require a change to the PROPATH or for run-time updates to be enabled.

The `refreshagents` API works by terminating all sessions within the agent of an ABL application, which causes the multi-session agent to refresh so that changes to the application can be picked up by a new agent. When you use this API, the ABL sessions that are running in the application complete before they are terminated. New sessions are started on the next session as needed, per the `minAvailableABLSessions` setting. The `refreshagents` API allows new sessions to use the updated persistent procedures, static objects, or online schema changes, and start running sessions against the new code. The following video introduces the refresh agents API in PAS for OpenEdge:

Refresh agents on a PAS for OpenEdge Instance

Enhanced ABL session management

In OpenEdge, the PAS for OpenEdge multi-session agent self-manages its ABL session pool to guarantee that an ABL Session is ready and available before it receives a request to process. The PAS for OpenEdge ability to self-manage supports continuous operations of ABL applications by addressing an unexpected failure due to resource unavailability. For example, creation of an ABL session may fail if the database it is connecting to runs out of capacity. The ABL session automatically reconnects to a session pool that has available resources. For more information, see Enhanced ABL session management.

Use a load balancer with OpenEdge

Modern, high-traffic applications must serve thousands of concurrent requests from users or clients, and return the correct content quickly and reliably. To cost-effectively scale to meet this high volume, modern computing best practice generally requires a load balancer.

Fault tolerance and load balancing can be achieved across multiple PAS for OpenEdge instances that support a single session-free business application by registering the instances with a Domain Name System (DNS) load balancing service like Elastic Load Balancer from AWS. In this case, your PAS for OpenEdge clients connect to the DNS load balancer using a URL that identifies the same OpenEdge ABL or SQL application and the appropriate connection transport supported by all the registered PAS for OpenEdge instances. Depending on the load balancing service, it can then delegate client requests to the available PAS for OpenEdge instances using several different load balancing policies.

Reasons to use a load balancer

There are many reasons for incorporating load balancers into your application's architecture. A load balancer can:

- Eliminate OS images and application servers as a SPOF
- Increase the number of clients concurrently using your application
- Update your application services with functionality that uses more OS image resources
- Update your application without schedule downtime
- Route end user access to application server instances located in multiple geographic locations
- Meet end user service requirements for application availability and response times

What is a load balancer?

A load balancer acts as a traffic director that sits in front of an enterprise's PAS for OpenEdge instances, and routes requests across all servers capable of fulfilling those requests, to maximize speed and capacity utilization. Using a load balancer ensures that no PAS for OpenEdge instance is overworked. If a single server goes down, then the load balancer redirects traffic to the remaining online PAS instances.

After you have decided that load balancing can improve your application availability, it becomes easier if you understand a few of the basics. When you begin investigating this area, you will discover many hardware and software load balancer products that offer different features and capabilities. As you view those features and capabilities understand that choosing a load balancer product is heavily influenced by your ABL application's design, implementation, and deployment. Choosing the correct load balancer product that matches your ABL application's runtime requirements is be crucial to maximize performance at the lowest cost.

There are several components you will find in every load balancer deployment:

- PAS for OpenEdge clients
- A 3rd party load balancer hardware or software product
- Multiple cloned instances of PAS for OpenEdge and your ABL application installed
- OpenEdge network connected database servers
- Monitoring tools for the load balancer, OS images, application servers, and database servers

Other more advanced features for certain load balancing products can be available:

- PAS for OpenEdge HealthScanner
- Autoscaling PAS for OpenEdge instances
- Regional traffic routing to PAS for OpenEdge instances
- Heart-beat tracking between load balancer instances for eliminating SPOF

How to choose a load balancer

Part of choosing the right load balancer product is related to the environment that your PAS for OpenEdge hosted ABL application is deployed. No single load balancer product is the right fit for every type of deployment environment. In most cases, software load balancer products are first on your list because they can work in most of the deployment environments. However, hardware load balancers can also be a candidate for certain types of deployments.

If your ABL application's PAS for OpenEdge instance is running in a container, then focus on the load balancer products that are tailored for working with swarms of containers.

If your deployment environment is in a cloud environment, then focus on the cloud's load balancer solution first as it is tailored for efficient interaction with cloud images and application servers. Then, look towards other software load balancer solutions that are designed for cloud operations.

For more information see, *ABL Application Continuous Operation* by OpenEdge Senior Architect Mike Jacobs.

V

Monitor

For details, see the following topics:

- Learn about monitoring
- Detect problems using OpenEdge HealthScanner
- Get insight with OpenEdge Management
- Get more information with deferred logging

Learn about monitoring

No application server and application will run forever. Leaks in OS processes, memory, files, and other shared resources will eventually require it to be restarted before it crashes. This in the past has been an expensive manual monitoring and OS administration evolution.

Now, more advanced load balancer products can include a ‘health check’ feature that provides automated checking and preemptive actions that avoid hard fail-over situations. The bulk of monitoring and OS image and application server administration available for you includes:

- Pinging PAS for OpenEdge and the OS image to verify their health and ability to continue servicing client requests and connections. The purpose is to identify a struggling application server or OS and permit it to be removed from service before it crashes and affects application data.
- Remove any single PAS for OpenEdge application instance, that fails its health check ping, from the pool of load balanced application servers before it crashes. Once removed from the pool and the application server becomes idle, it may be stopped, maintained, and restarted.

Monitoring includes the OS images in your application architecture. OS images should be monitored for performance and health (CPU, Memory, SWAP, Disk space, Disk I/O) and errors in system logs. Networks should be monitored for collisions, retransmissions, routing, latency, and other performance issues, not just from the outside but between machines internally. PAS for OpenEdge instances should monitored for errors, memory leaks, and client response time performance. There are a multitude of monitoring values for PAS for OpenEdge using REST API or JMX beans. The OpenEdge HealthScanner can monitor your PAS for OpenEdge instance health and work with a load balancer's health check. Databases should be monitored for performance, replication performance, and overall health. Logs should be monitored for errors and warnings.

Detect problems using OpenEdge HealthScanner

The OpenEdge HealthScanner is an essential monitoring tool for maintaining continuous operation of your application's ecosystem.

This tool monitors system and server performance and can be used to automate recycling agent processes to reduce failure risks and increase application uptime. Designed for system administrators, OpenEdge HealthScanner scans the system continuously, to monitor key health metrics, and when called, indicates if pre-set thresholds are exceeded. For the continuous operation solution, the HealthScanner is used in tandem with a load balancer to identify and remove problematic application server instances before they fail. This preemptive action saves an enterprise and its users from experiencing application downtime.

For more information, see 'Use the OpenEdge HealthScanner' in *Manage Progress Application Server (PAS) for OpenEdge*.

Get insight with OpenEdge Management

A large part of maintaining a continuous operation is having a way to extensively monitor your application architecture, and being able to take preemptive or retroactive action when a problem arises. OpenEdge Management is a tool that you can use to monitor your systems to run your application continuously.

OpenEdge Management is a powerful tool that you can use monitor, alert, and trend almost every facet of your application architecture. OpenEdge Management can easily be extended to do even more monitoring and take corrective action when an alert occurs by using its programmatic capabilities. For continuous operation, you can use OpenEdge Management to monitor your network resources, your local and remote systems, file, database, and PAS for OpenEdge resources. OpenEdge Management provides feedback about each resource it monitors based on performance criteria that you establish. If a resource's performance does not meet the criteria you set, then you determine the action OpenEdge Management takes. Those actions can be as simple as sending an email or in-application alert, or as complex as creating a new instance of PAS for OpenEdge.

While your application architecture might be made up of several systems and components, there are a number of resources that you should monitor to ensure that your application and system runs continuously. When you are determining which resource to monitor, think of each critical system and component that act as linchpins for your architecture. In general, you should monitor key resources like:

- Your database
- Your system resources, such as CPU, memory, or disk
- A network resource, such as HTTP communications
- A file resource, such as a log file
- Your PAS for OpenEdge instances

In OpenEdge Management, you select the resource you want to monitor, and create a monitoring plan for those resources.

Main steps: Monitor your resources

1. Create a monitoring plan.
2. Set rules.
3. Configure alerts.
4. Create a collection.
5. Create trending reports.

For more information, see.Resource monitoring with OpenEdge Management.

Create a monitoring plan

When you create a monitoring plan, you provide both a monitoring definition and one or more rule definitions.

The monitoring definition includes:

- The schedule selected from all the schedules that are available. The schedule specifies when the plan is active, and the available schedules include only those schedules that do not overlap with schedules already in use for a resource.
- A polling interval, which indicates the frequency with which this resource's rules are checked by OpenEdge Management.
- An option to enable alerts for this resource. Enabling alerts means that alerts are generated while the plan is active.
- An option to store, or trend, the performance data collected by this resource to the OpenEdge Management Trend Database while the plan is active.

The number of polls after which performance data is trended to the OpenEdge Management Trend Database.

The rule definition identifies:

- When to generate an alert for the resource
- What level of alert severity cause OpenEdge Management to generate an alert
- The number of failed polls after OpenEdge Management generates an alert
- The action that OpenEdge Management performs when an alert is generated
- The number of successful polls after OpenEdge Management clears an alert
- The action OpenEdge Management performs when an alert is cleared

A schedule defines a block of time for which a set of monitoring rules is active for a resource. When you add a monitoring plan to a resource, you specify the schedule to indicate when the monitoring plan is active.

OpenEdge Management provides several predefined schedules. You can use the schedules as they are, or you can modify them to suit your operating needs. You can also use an existing schedule as the basis for a new schedule by changing its name and characteristics.

Set rules for continuous operation

Rule definitions identify the specific attributes of a resource that you want to monitor, the severity of an alert OpenEdge Management generates in response to performance, and the action that occurs when an alert is generated. You set specific rules for each resource type. For example, the rule definition for a disk resource includes setting an alert that triggers when disk activity exceeds a specified percentage.

For more information, see:

- Monitor system resources in *OpenEdge Management: Resource Monitoring*
- Monitor network resources in *OpenEdge Management: Resource Monitoring*
- Monitor file resources in *OpenEdge Management: Resource Monitoring*

Configure alerts for continuous operation

An alert is an automatic, user-defined process that occurs in response to a rule violation on a monitored resource. You create actions and associate them with rules to ensure that certain activities occur when the rule triggers. You use these actions to send e-mails, record the occurrence of an alert in a log file, send an Simple Network Management Protocol (SNMP) trap, or perform a combination of these activities.

For more information, see Alerts in OpenEdge Management.

Create a collection

In OpenEdge Management, you can create one or more groups of resources, known as collections. When you create a collection, you typically group resources together that have a common element, like all the database resources used by one division in your company or all the resources that support one particular application. When you group resources, you can more easily examine their details and assess their status.

For example, you might create a collection known as Collection A that includes all resources that a particular application depends on. Because you have more than one application, you then create another collection known as Collection B, which is also dependent on one of the resources in Collection A. The common resource to both Collection A and Collection B in this example is disk space. Within these two collections, you can set an alert to notify you of rapidly filling disk space. When the alert fires, you are notified that both collections, and by extension, both applications, are at risk. This notification enables you to quickly determine the extent and the effect of a potential failure and to take action to avoid that failure.

For more information, see Create a collection.

Create trending reports

You can choose to have OpenEdge Management trend performance data to the OpenEdge Management Trend Database. Different trend expectations exist for different resource types. Therefore, each trend interval has a different meaning depending on the data to be trended.

If you set a trending value in the **Trend Performance Data every** field, OpenEdge Management uses the value you provide in combination with the value in the **Polling Interval** field to calculate the trending interval. For example, if you initially the polling interval for 30 seconds and also set the trending value to trend data every 3 polls, OpenEdge Management will automatically calculate the trending interval as 90 seconds. You will see this result when you view the summary of the resource monitor's performance. If performance falters, then you can take corrective action before the problem that is causing performance issues becomes catastrophic.

For more information, see Understand reports.

Get more information with deferred logging

If one of your PAS for OpenEdge instances does go down, then a new instance can be generated by your load balancer. However, to determine whether an agent crash can be prepared for in the future, deferred logging helps system administrators understand the cause of an agent crash by providing additional logging information.

Deferred logging also helps performance by only recording a small amount of information that is relevant to an agent crash.

Deferred logging in PAS for OpenEdge opens an in-memory buffer to record information immediately preceding a multi-session agent crash. Deferred logging also lets system administrators run an on-demand monitoring check on the instance. The in-memory buffer is configured independently from the agent log and automatically writes to the agent log file in the event of a crash or when it is called through an API.

For more information, see 'Use deferred logging in PAS for OpenEdge' in *Manage Progress Application Server (PAS) for OpenEdge*.

VI

Tools

For details, see the following topics:

- DSRUTIL
- PROMON utility
- Database logs

DSRUTIL

The DSRUTIL command-line utility enables database administrators to validate the properties of replication-enabled source and target databases as well as confirm the status of target database brokers. With this feature, you can:

- More readily identify databases within your replication environment
- Determine the operational status of brokers within your replication environment

For more information, see OpenEdge Replication DSRUTIL Properties Validation Utility.

PROMON utility

The PROMON utility lets you disconnect secondary brokers without issuing a process kill signal. Secondary brokers let you manage the configuration for a specific group of end users, such as ABL clients, SQL clients, or both.

This option reduces potential database downtime. For more information, see 'Server Options' in *Manage the OpenEdge Database*.

To support continuous operation, the server's ID number and the server's user ID are now stored as separate fields, `Sv No` and `SvUsr No`. These fields, which apply only to remote clients and display on:

- The **User Control** screen
- The **Servers** and **Servers by broker** screens in R&D Status Displays
- The **Terminate a Server/Secondary Broker** screen in R&D Administrative Functions

Database logs

The database log (`.lg`) captures information that helps you understand the circumstances surrounding a failure. The log now records startup parameters for secondary brokers.

Although you can list broker parameters for running servers by using PROMON R&D Status or Virtual System Tables (VTS), the database log file lets you troubleshoot landscape or connection issues after a server failure. You can easily determine which brokers were active during an event and how they were configured when they started.

For more information about database logs, see 'OpenEdge database log file' in *Manage the OpenEdge Database*.

VII

Learn more

For details, see the following topics:

- Related solutions

Related solutions

Other solution guides that you may find useful include:

- Learn about modernization in OpenEdge
- Learn about OpenEdge performance
- Learn about CI/CD