Progress®



OpenEdge 12.4 Release Enablement

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

Last updated with new content: Release 12.4

Updated: 2021/06/28

Table of Contents

OpenEdge 12.4 Release Enablement.....7

Language.....8
- TLS client authentication from an ABL socket.....8
- TLS client authentication from an ABL HTTP client.....10
- Default area assignments for database objects.....11
- Online schema changes - Field drop from ABL.....12
- Server-side single table aggregation support.....13
- Cancel SQL queries via JDBC clients.....14
- Methods as callbacks for ABL SAX-READER objects.....14
- Methods as callbacks for SOAP web service.....15
- VAR statement enhancements.....16
- Open Client support for .NET Core and .NET Framework.....19
- Online schema support for delta.df in Data Dictionary.....19

Database.....20
- TLS support for OpenEdge database replication.....20
- Ability to change the database master key online.....21
- Online table truncation 25
- Database maintenance mode.....26
- Synchronize database replication source with target databases.....27

PAS for OpenEdge.....30
- ABL Service Deployment Packaging.....30
- Enhanced troubleshooting for PAS for OpenEdge multi-session agent.....30

OpenEdge Command Center.....31

1

OpenEdge 12.4 Release Enablement



OpenEdge 12.4 delivers updates and enhancements to the following product areas:

- Language on page 8
- Database on page 20
- PAS for OpenEdge on page 30

The release also introduces:

- OpenEdge Command Center on page 31

For details, see the following topics:

- Language
- Database
- PAS for OpenEdge
- OpenEdge Command Center

Language

TLS client authentication from an ABL socket

PAS for OpenEdge supports an ABL client socket that uses client authentication for a TLS connection to any OpenEdge or non-OpenEdge server. This feature allows application developers access to web services or servers that require client authentication to meet enterprise security policy requirements.

An ABL socket supports client authentication to the following servers:

- Local PAS for OpenEdge instances
- PAS for OpenEdge instances running on Amazon web services
- PAS for OpenEdge instances that use an HTTP load balancer
- Apache HTTP servers (accessed either locally or through a VPN)
- Apache Tomcat servers (accessed either locally or through a VPN)
- Jetty servers (accessed either locally or through a VPN)
- Internet Information Services (IIS) servers (accessed either locally or through a VPN)
- Client- and server-side HTTP Proxy servers

The following video explains how an ABL client socket is used for client authentication.

See a video on TLS Client Authentication from an ABL Socket here:
https://www.youtube.com/watch?v=qYTtdHOxDeM

For more information, see 'CONNECT( ) method (Socket object)' in *ABL Reference*.

How to use TLS client authentication from an ABL Socket

Use the following `CONNECT( )` to support client authentication from an ABL socket:

- **`-SSLAuth`**—Specifies if client authentication is required.
- **`-SSLKeyFile`**—Specifies the location of the client certificate.
- **`-SSLKeyPwd`**—Specifies whether the password for the client certificate is in clear text or encoded format.

In previous releases of OpenEdge, you could only use these parameters with classic AppServer instances using SOAP web services, In this release, the code is enhanced to support all client authentication. The input parameters are also used when establishing a TCP connection.

The connection parameters are specific to the ABL socket handle and are only used for client authentication, The parameters do not affect any other existing TLS options.

The following is an example of `CREATE SOCKET hSocket`:

```
hSocket: CONNECT ("-H localhost -S 27488 -ssl -sslAuth ssl
      -sslKeyFile PAS-default-cert.pem -sslKeyPwd password")
```

Troubleshooting

The troubleshooting information in this section is based on a scenario where an ABL client connects to a PAS for OpenEdge instance that is configured to use the client authentication. The default protocol used for the handshake between the client and server is TLS v1.2.

To ensure you receive adequate troubleshooting information, ensure that both the client and server are configured to log TLS handshake messages. For more information, see the following Knowledge Base article: How to enable SSL debugging in OpenEdge?

The following errors can occur in this described scenario.

Use Case: Client authentication is enabled on the server but the client does not supply the required credentials

The following server message appears:

```
....
javax.net.ssl|DEBUG|1C|thd-4|2021-06-10 02:44:58.262 EDT|
CertificateMessage.java:364|Consuming client Certificate handshake message (
"Certificates": <empty list>

)
javax.net.ssl|ERROR|1C|thd-4|2021-06-10 02:44:58.263 EDT|TransportContext.java:313|
Fatal (BAD_CERTIFICATE): Empty server certificate chain
```

The following client messages appear:

```
....
INTERNAL STATE OPERATION – read:fatal:bad certificate
INTERNAL STATE OPERATION – SSL_connect:error in error
ID-0x02bb04c0 CTX-0x02bda250 BIO-0x02be28d0 ERROR – SSL Client handshake failure
(336151570) SSL routines
ID-0x02bb04c0 CTX-0x02bda250 BIO-0x02be28d0 INFO – Terminated SSL Client session
```

The following error appears in the ABL client output:

```
Secure Socket Layer (SSL) failure. error code 336151570: SSL routines (9318)
Connection failure for host localhost port 28986 transport HTTPS. (9407)
```

Use Case: Client authentication is enabled on the server, but no password is supplied by the client

In this scenario, the handshake between the client and server can't be established.

The following error appears in the ABL client output:

```
Secure Socket Layer (SSL) failure. error code 0: Unknown network error (9318)
Unknown transport error received 0. (5490)
```

Use Case: Client authentication is enabled on the server but the client supplies the wrong password

The following server message appears:

```
....
javax.net.ssl|DEBUG|1C|thd-3|2021-06-10 03:30:47.608 EDT|ServerHelloDone.java:97|
Produced ServerHelloDone handshake message (
   <empty>
  )
```

The following client messages appear:

```
....
INTERNAL STATE OPERATION – SSL_connect:error in /TLS read server certificate request
ID-0x01f057c0 CTX-0x01f085e0 BIO-0x01f35e20 ERROR – SSL Client handshake failure
(101077092)
digital envelope routines
ID-0x01f057c0 CTX-0x01f085e0 BIO-0x01f35e20 INFO – Terminated SSL Client session
```

The following error appears in the ABL client output:

```
Secure Socket Layer (SSL) failure. error code 101077092: digital envelope routines (9318)
Connection failure for host localhost port 28986 transport HTTPS. (9407)
```

Use Case: Client authentication is successful

The following server message appears:

```
....
javax.net.ssl|DEBUG|1A|thd-2|2021-06-14 06:52:37.966 EDT|Finished.java:581|
Consuming client Finished handshake message (
   "Finished": {
     "verify data": {
    0000: D3 C6 A4 9B 95 F9 48 12   0D 13 02 97
     }'}
   )
   javax.net.ssl|DEBUG|1A|thd-2|2021-06-14 06:52:37.967 EDT|Finished.java:450|
Produced server Finished handshake message (
   "Finished": {
     "verify data": {
    0000: F6 C7 53 37 95 BE DD 93   C9 E1 5B 8C
     }'}
   )
```

The following client messages appear:

```
....
INTERNAL STATE OPERATION – SSL State: 32 SSL negotiation finished successfully
ID-0x01cef7c0 CTX-0x01cf2c80 BIO-0x01cf4930 INFO – SSL Client handshake succeeded with
host (localhost)
```

No errors appear in the ABL client output.

TLS client authentication from an ABL HTTP client

The `OpenEdge.Net` procedure library now supports client-side TLS authentication, in addition to the server-side authentication that was available in earlier OpenEdge versions.

Application developers can use the `OpenEdge.Net` library to write ABL HTTP client applications that verify a server certificate and also send a client certificate to any TLS-enabled web server, including PAS for OpenEdge instances and non-OpenEdge servers such as Apache HTTP Server and Microsoft IIS.

This feature enables OpenEdge application developers to keep their ABL HTTP client applications up to date with enhanced security policies.

The following video demonstrates how to use an ABL HTTP client application to send a client TLS certificate to a server.

See a demonstration of TLS client authentication in an ABL HTTP client here:

For more information, see 'Configure TLS security settings' in *Develop an HTTP Client Application in ABL*.

How to implement TLS client authentication in an ABL HTTP client

The ABL HTTP client requires access to a PEM file that contains both the public and the private key of the client. Once you have a PEM file ready, perform the following steps in your ABL HTTP client code:

1. Create a `TlsClientCredentials` object containing two properties:

- A path to the PEM file. You can specify either an absolute path or a path relative to a directory listed in the `PROPATH`.

- An encoded passphrase to enable the HTTP client to access the private key in the PEM file.

```
var EncodedString oEncSt.
var TlsClientCredentials oCred.

// Create a passphrase for the client certificate using plain-text encoding.
oEncSt = new EncodedString().
oEncSt:encoding = EncodingTypeEnum:nopr0.
oEncSt:value = "password".

// Create credentials which consists of certificate path and encoded passphrase.
oCred = new TlsClientCredentials().

oCred:identity = new OpenEdge.Core.String("PAS-default-cert.pem").

oCred:secret = oEncSt.
```

2. Pass the `TlsClientCredentials` object to the `UsingClientCredentials()` method during the construction of the HTTP request object.

```
...
// Create credentials which consists of certificate path and encoded passphrase.
oCred = new TlsClientCredentials().
oCred:identity = new
OpenEdge.Core.String("C:\OpenEdge\WRK\oepas1\bin\PAS-default-cert.pem").
oCred:secret = oEncSt.

// Create the request with stated credentials, calling the built-in REST ping service.
oRequest = RequestBuilder:Get("https://httpbin.org/get")
                         :UsingClientCredentials(oCred)
                         :Request.
```

You can also use a callback to obtain a TLS client credentials object. Callbacks are useful when you need to use different client certificates—for example, if you have multiple users connecting to the server and each user has their own client certificate. In such situations, you should delegate the task of constructing the `TlsClientCredentials` object to an ABL handler class and then use a callback from within the ABL HTTP client code.

For more information about callbacks in TLS client authentication, see 'Configure TLS security settings' in *Develop an HTTP Client Application in ABL*.

Troubleshooting

The `OpenEdge.Net` procedure library internally uses ABL client sockets to establish TLS connections with servers. To know more about troubleshooting TLS-related problems in an ABL HTTP client, see the troubleshooting information in the section titled TLS client authentication from an ABL socket on page 8.

Default area assignments for database objects

By default, ABL and SQL create new data objects in the schema area, and the Data Dictionary creates new database objects in the next storage area after the schema area. To take advantage of the performance capabilities of Type II areas, database administrators can specify a default area for the creation of tables, LOBs, and indexes, so that objects are created in a Type II area instead of a Type I area. You can also change the default object creation area, whether or not the database is running. Default area assignments eliminate downtime when moving tables, LOBs, and indexes from a Type I to a Type II storage area.

You can now use the PROUTIL utility to configure or display default areas.

Note:

For an overview of storage areas, see 'Storage areas' in *OpenEdge Database Essentials*.

This video demonstrates the PROUTIL commands and Data Dictionary functionality for using default storage areas for objects. After demonstrating commands, the speaker describes two use cases for default storage areas.

See a video on default area assignments here: https://www.youtube.com/watch?v=r6Nt6kJSrdo

How to use default storage areas for database objects

For details, see:

- 'Set default storage areas for creating database objects' in *Manage the OpenEdge Database*
- 'Show default storage areas for creating database objects' in *Manage the OpenEdge Database*
- 'Create objects in default areas' in *OpenEdge Data Management: SQL Development*
- 'Work with default areas' in *OpenEdge Data Management: SQL Development*

Troubleshooting

The `DBRESTRICTIONS` function in ABL is enhanced to return the keyword `DEFAULT-AREAS` if the specified database cannot use default areas. For details, see 'DBRESTRICTIONS function' in *ABL Reference*.

Make sure that you create default areas in valid areas.

If your create or alter operation fails, check for errors like the following:

Area type	ABL message	SQL message
Invalid	`Can't get area attributes for area 55 (16493). Area 55 is not valid as the default aea of tables.`	`Database storage area "Invalid Area" cannot be found in db schema.`
Type I	`Area 6 is not valid as the default area for table (20223).`	`Database storage area "MyTypeI" is required to be a Type II storage area, but it is not.`
Encryption	`Area 80 is not valid as the default area for table (20223).`	`Invalid area number specified.`

Online schema changes - Field drop from ABL

You can now remove a named field from the schema definition while still being connected to the database by means of the online `Drop Field` transaction. The data of the field in the database table is also effectively removed in the process without requiring any exclusive lock.

This video shows how to use the `Drop Field` functionality while the database is still running.

See a video on the online `Drop Field` functionality here: https://www.youtube.com/watch?v=OZhgQD9tIJg

How to use online schema changes - Field drop from ABL

You can drop a field online using the Data Adminstration tool.

For more information, see 'Drop a field online' in *Optimize the OpenEdge Database for Continuous Operation>*.

Troubleshooting

- If `Drop Field` is part of a transaction with other schema changes, then the entire transaction, including `Drop Field`, occurs offline.
- If the `dbnotification` feature of the database is turned off, then `Drop Field` can only be offline.
- In a single transaction, you can drop only one field online.
- You cannot drop an LOB field online.
- Online field drop works only if the database and all the clients are of version 12.4 or above.

Server-side single table aggregation support

This release of OpenEdge introduces a new ABL `AGGREGATE` statement for computing aggregate values (summary information about a result set) on the database server. Previously, aggregate calculations were performed client-side using both the `ACCUMULATE` statement and the `ACCUM` function. Now, ABL application developers can simply use a single `AGGREGATE` statement for `COUNT`, `TOTAL`, and `AVERAGE` aggregate operations. The `AGGREGATE` statement simplifies coding and improves performance as the computation is done efficiently on the database server.

The following code example shows an `AGGREGATE` statement being used to calculate the average balance for customers in Boston:

```
VAR DECIMAL avgBalance.
AGGREGATE avgBalance = AVERAGE(Balance) FOR Customer
  WHERE Country EQ 'USA' AND City EQ 'Boston'.

MESSAGE "Average balance: " avgBalance
  VIEW-AS ALERT-BOX.
```

This video demonstrates support for server-side single table aggregation with the `COUNT`, `AVERAGE`, and `TOTAL` operations.

See a video on server-side aggregation support here: https://www.youtube.com/watch?v=xupgswx60jI

How to use the AGGREGATE statement

For more information, see 'AGGREGATE statement' in *ABL Reference*.

Troubleshooting

- The `AGGREGATE` statement can only be used with a single aggregate on a single table.
- The `AGGREGATE` statement is not allowed for DataServers.
- The `AGGREGATE` statement cannot be inside a `FOR` loop for the same buffer.
- If the session has a record in scope, then that record is no longer in scope after the `AGGREGATE` statement executes. Therefore, you should not use the same buffers in `AGGREGATE` statements that are used in other parts of the application.
- The `AGGREGATE` statement is not supported with the `TENANT-WHERE` clause for multi-tenant tables.

Cancel SQL queries via JDBC clients

You can now use the JDBC Cancel() API to terminate a SQL query in a JDBC client. When the JDBC client executes the API, the OpenEdge SQL server cancels the query and discards all running operations for that query without impacting other queries or applications. The JDBC Cancel() API can be used to cancel SQL SELECT queries that are taking longer than expected.

This video demonstrates support for the Cancel() API SQL JDBC clients.

See a video on Cancel SQL queries via JDBC clients here: https://www.youtube.com/watch?v=mnvlbZN_Xr8

Troubleshooting

- Only SELECT statements can be cancelled. Executing the Cancel() API on other statements will be ignored.
- In order for the Cancel() API to work, the client must use the latest OpenEdge JDBC driver.

Methods as callbacks for ABL SAX-READER objects

This feature allows ABL developers to register class methods as event handlers for SAX-READER events. This capability allows ABL developers to use an object-oriented ABL solution to handle callback events within their applications and avoid mixing procedural and object oriented ABL code.

To enable this feature, add the ABL EVENT-HANDLER-OBJECT attribute to the handle attributes for the SAX-Reader object. The EVENT-HANDLER-OBJECT accepts object-oriented ABL objects as the callback handler events generated by the ABL SAX-Reader object handle.

New ABL callback method signatures introduced for this feature are the same as those defined in the SAX2 documentation at www.saxproject.org.

See a video on methods as callbacks for ABL SAX-READER objects here: https://www.youtube.com/watch?v=RpaCpbUapHE

How to use Methods as callbacks for ABL SAX-READER objects

To use this feature:

1. Define the callback methods in an object-oriented ABL class (.cls) file. For more information on SAX-Reader event callbacks method signatures and syntax, see 'SAX callback reference' in *Use XML with ABL Applications*.
2. Create an object instance of the class in your application and set the SAX-Reader EVENTHANDLER-OBJECT attribute to the ABL object referenced as the callback handle for the SAX-Reader event. For more information, see 'ABL SAX callbacks' in *Use XML with ABL Applications*.

For a detailed example with class definitions and sample code, see 'Read an XML file using methods as callback' in *Use XML with ABL Applications*.

Troubleshooting

This section describes common error scenarios.

Use Case: Callback object is deleted before the callback assignment

In the this scenario, the AVM logs the following error:

```
EVENT-HANDLER-OBJECT attribute value must have a valid object referene. (20074)
```

To fix this error, identify the code segment that deletes the callback object before the object reference is assigned to the EVENT-HANDLER-OBJECT attribute of the SAX-READER object. Move the Delete statement to an appropriate location in the code (usually after all the XML documents are read and processed by the SAX-READER object).

Use Case: Callback object is deleted after the callback assignment

In this scenario, the AVM logs the following error:

```
Unable to set attribute EVENT-HANDLER-OBJECT in widget of type SAX-READER. (3131)
```

Identify the code segment that deletes the callback object after the object reference is assigned to the EVENT-HANDLER-OBJECT attribute of the SAX-READER object, but before the SAX-READER object starts processing the XML document. Move the Delete statement to an appropriate location in the code (usually after all the XML documents are read and processed by the SAX-READER object).

Use Case: Callback events are not getting executed

In this scenario, no errors are generated. The SAX-READER object parses the XML document and since there are no matching callback methods to execute, no callback methods are executed.

To resolve this issue:

- Ensure the correct callback object is assigned to the EVENT-HANDLER-OBJECT attribute of the SAX-READER object.
- Ensure the signature of the callback methods match the prescribed syntax. For more more information on the callbacks methods syntax, see 'SAX callback reference' in *Use XML with ABL Applications*.

Methods as callbacks for SOAP web service

This feature allows ABL developers to register class methods as SOAP header callback targets for SOAP web service operations. It also provides the capability to register class methods as event handlers to use built-in ABL functionality to run SOAP web service operations asynchronously.

To enable this feature, the following ABL entities and constructs were extended:

- The SET-CALLBACK( ) method can now set methods as callback handlers for SOAP events.
- The EVENT-HANDLER attribute in the RUN ... ASYNCHRONOUS statement is extended to set methods as callback handlers for asynchronous SOAP web service operations.

This video introduces methods as callbacks for SOAP web service.

See a video on methods as callbacks for SOAP web services here:
https://www.youtube.com/watch?v=Wy8nRPxSN_g

For more details, see:

- 'SET-CALLBACK( ) method' in *ABL Reference*
- 'Run statement' in *ABL Reference*
- 'Statements to consume SOAP web services' in *Develop an ABL SOAP Client*
- 'Attributes, methods, and events to consume web services' in *Develop an ABL SOAP Client*
- 'Handles to consume a web service' in *Develop an ABL SOAP Client*

How to use methods as callbacks for SOAP web service

To use the methods as callbacks for request and response SOAP events:

- Create an instance of the class containing the callback methods.
- Use the SET-CALLBACK( ) method with appropriate parameters for the request and response headers. For more information, see 'SET-CALLBACK( ) method' in *ABL Reference*.

Troubleshooting

This section describes common error scenarios.

Use Case: Callback object is deleted before it is set as a callback object

The AVM logs the following error:

```
Third argument to SET-CALLBACK must be a valid procedure handle or object reference. (13273)
```

Identify the code segment that deletes the callback object before it is set using SET-CALLBACK and move the Delete statement to an appropriate location in the code (usually after all the SOAP requests and response processing is complete).

Use Case: Callback object is deleted after it is set as a callback object

This error generates a quit condition and the AVM stops. The following error is recorded:

```
Invalid event-handler-context encountered when trying to invoke event handler 'ProcessCallback'. (20055)
```

Identify the code segment that deletes the callback object after it is set using SET-CALLBACK and before the SOAP request and response processing is complete. Ensure you delete the callback object after all SOAP service calls are completed.

Use Case: SET-CALLBACK is addressing an invalid callback signature

The AVM logs the following error:

- `Invalid handle. Not initialized or points to a deleted object. (3135)`
- `Cannot execute WAIT-FOR statement when a invalid widget handle is encountered. (4012)`

Ensure the method signatures for the callback match the method signatures you defined in the callback object. The error also occurs if you are pointing to a method in the callback object that does not exist.

VAR statement enhancements

In this release of OpenEdge, the VAR statement is enhanced in several ways that allow ABL application developers to:

- Instantiate and initialize class variables, including array object variables
- Use expressions during variable initialization, including MEMPTR, RAW, and ROWID variables

These features simplify coding, allowing developers to define program variables more concisely, in a manner similar to other programming languages.

For example:

```
/* Instantiate and initialize a class variable */
VAR myclass myobj = NEW myclass("Progress", 2021, ?).

/* Instantiate and initialize array object variable */
VAR StateClass[] objArrayB = [NEW StateClass(“MA”), NEW StateClass(“NH”)].

/* Use expressions during initialization */
VAR INT[3] x = [funct( ), 123, a + b].

/* Initialize RAW variable using expressions */
VAR RAW[3] customerByteInfo =
[RAW(Customer.Name),RAW(Customer.Address),RAW(Customer.City)].
```

This video introduces the VAR statement functionality in both the OpenEdge 12.3 and 12.4 versions.

See a video on the 12.3 and 12.4 enhancements to the VAR statement here:
https://www.youtube.com/watch?v=NW9_0wjT4hA

How to use the VAR statement

For more information, see:

- 'Learn about the VAR statement' in *Develop ABL Applications*
- 'VAR statement' in *ABL Reference*

Troubleshooting

The VAR statement is very intuitive, but there are some considerations to be aware of:

- The VAR statement does not allow data type abbreviations, unlike the DEFINE VARIABLE statement. New keywords were introduced in OpenEdge 12.3: CHAR (for CHARACTER) and INT (for INTEGER). Therefore, CHAR and INT may be specified when using the VAR statement, but all other data types must be fully spelled out.
- Other options allowed with the DEFINE VARIABLE statement, such as FORMAT or LABEL, cannot be used with VAR. If you need those other options, use the DEFINE VARIABLE statement instead.
- VAR may only be used at the beginning of an ABL routine, before any executable statements in the block. If the placement of the VAR statement does not satisfy this criteria, you see compilation error 20006.
- Uninitialized elements of a **scalar** array variable inherit the value of the last initialized element. This behavior is the same as the DEFINE VARIABLE statement.

  ```
  VAR INT[3] x = [1,2]. //3rd element defaults to 2.
  ```

- Uninitialized elements in a determinate array **object** variable are set to the Unknown (?) value.

  ```
  VAR StateClass[2] objArrayA = [NEW StateClass(“MA”)]. //2nd element is set to ?
  ```

- If you define a variable using VAR, and use an expression to set the initial values, and then later use a DEFINE VARIABLE statement with the LIKE option on that variable, then the new variable only inherits the data type and extent. The initial values are not inherited since they came from an expression and the compiler won't know what they are.

  ```
  VAR INTEGER[] ivar = [ix1,ix2,ix3].
  DEFINE VARIABLE ix LIKE ivar.  //inherits type and extent but not initial values
  ```

- Although you can use VAR with a constant initial value prior to a Super( ) statement, you cannot use it with an expression as the initial value. This is invalid because at runtime, an assignment statement is created to handle the expression which no longer satisfies the rule for Super() (no executable statements are allowed prior to Super( )).

  ```
  CONSTRUCTOR PUBLIC StateClass (INPUT myChar AS CHARACTER ):
      VARIABLE INTEGER iLength = 0.  // valid syntax
      SUPER ().
  END CONSTRUCTOR.


  CONSTRUCTOR PUBLIC StateClass (INPUT myChar AS CHARACTER ):
      VARIABLE INTEGER iLength = iInstancePrivate.  // invalid syntax
      SUPER ().
  END CONSTRUCTOR.
  ```

- Using the VAR statement, you can define and initialize MEMPTR, RAW, and ROWID variables using an expression. This capability does not exist with the DEFINE VARIABLE statement.

Open Client support for .NET Core and .NET Framework

Development on .NET Framework has stopped and moved to .NET Core with the open source community. Because of this, OpenEdge has expanded Open Client support for .NET Core and .NET Framework so customers can continue to use .NET going forward.

Open Client for .NET supports .NET framework 4.8, .NET Core 3.1, and .NET Core 5.0. Open Client applications can be run on .NET platforms that support .NET standard 2.0 targeted libraries. Progress also supports running Open Client application on Linux .NET core 3.1, Windows .NET core 3.1, and .NET framework 4.8 on Windows.

See a video on Open Client support for .NET Core and .NET Framework here: https://www.youtube.com/watch?v=CyiYFJxjFWw

See a video demo on Open Client support for .NET Core and .NET Framework here: https://www.youtube.com/watch?v=jfIfFVZywoY

For more information, see 'Runtime support with .NET Core FAQ' in *OpenEdge Frequently Asked Questions*.

How to use Open Client support for .NET Core and .NET Framework

If your .NET application is targeting the .NET Framework, you do not have to change anything. You can update your app by including library references to the updated Open Client libraries and any generated proxy libraries.

If your .NET application is targeting .NET Core, then you will need to ensure your project has a package reference to the `System.Configuration.ConfigurationManager` version 4.7. You can add this package reference to your applications `csproj` file from within Microsoft Visual Studio or the .NET software development kit. Download the required package here.

Troubleshooting

Online schema support for delta.df in Data Dictionary

The Data Administration tool connects the two databases, compares the schemas, and generates a `.df` file that contains the changes required in order for the first database schema to match the second database schema. This is referred to as the `delta.df`.

The Data Administration tool does not help in distinguishing between online and offline operations. With the new feature, a new `delta.df` format is generated which consists of the following four sections:

- PreDeploy
- Trigger
- PostDeploy
- Offline

See the problem statement and the customer value video on online schema support for delta.df in Data Dictionary here: https://www.youtube.com/watch?v=01Kngl2Po-Y

See a demo on online schema support for delta.df in Data Dictionary here: https://www.youtube.com/watch?v=4-CVX8fBJdM

For more information, see 'Dump incremental delta.df supporting online schema changes' in *OpenEdge Database Tools*.

How to use online schema support for delta.df in Data Dictionary

The following sections are now available in the `delta.df` file. The sections explain the operations that are performed before and after the deployment of an application:

Section name	Description
PreDeploy	Contains all non-disruptive online operations and items such as: • Allow the field data type to change from **Integer** to **Int64**. • Modify the existing non-structural schema fields. • Rename a field in an existing table.
Trigger	Contains all the trigger definition changes such as add, delete, and modify. The database triggers have r-code and are deployed before the trigger schema changes. The trigger r-code is executed after the PreDeploy changes are made.
PostDeploy	Contains all online disruptive operations. These schema change operations are applied to the production system after confirming that the objects are not used in any other version of the running application.
Offline	Contains all schema change operations. These schema change operations cannot be carried out online and therefore require the following exclusive schema locks on the database: • Drop an inactive index from an existing table. • Drop a non-indexed field from an existing table.

Database

TLS support for OpenEdge database replication

System administrators need secure communication to protect data being transferred when OpenEdge Replication is deployed. Before this release of OpenEdge, the only way to use secure communication was to enable AI encryption, which does not encrypt all the data used by replication and is only available to customers with Enterprise licenses.

OpenSSL is a toolkit for the Transport Layer Security (TLS) and Secure Sockets Layer (SSL) protocols. In this release, OpenSSL support that was added to OpenEdge Replication provides additional security for message authentication for all OpenEdge Replication users and may save money related to security support calls.

See a video on TLS Support for OpenEdge Replication here: https://www.youtube.com/watch?v=2HbjtrOU0T0

For more information, see 'Transport Layer Security' in *Manage the OpenEdge Database*.

How to use TLS support for OpenEdge database replication

For details, see:

- 'Set up OpenEdge Replication for TLS' in *Manage the OpenEdge Database*

- 'Log TLS messages during OpenEdge Replication' in *Manage the OpenEdge Database*
- 'Monitor TLS for OpenEdge Replication' in *Manage the OpenEdge Database*

Troubleshooting

For more troubleshooting information, see 'Troubleshoot TLS issues for OpenEdge Replication' in *Manage the OpenEdge Database*.

Ability to change the database master key online

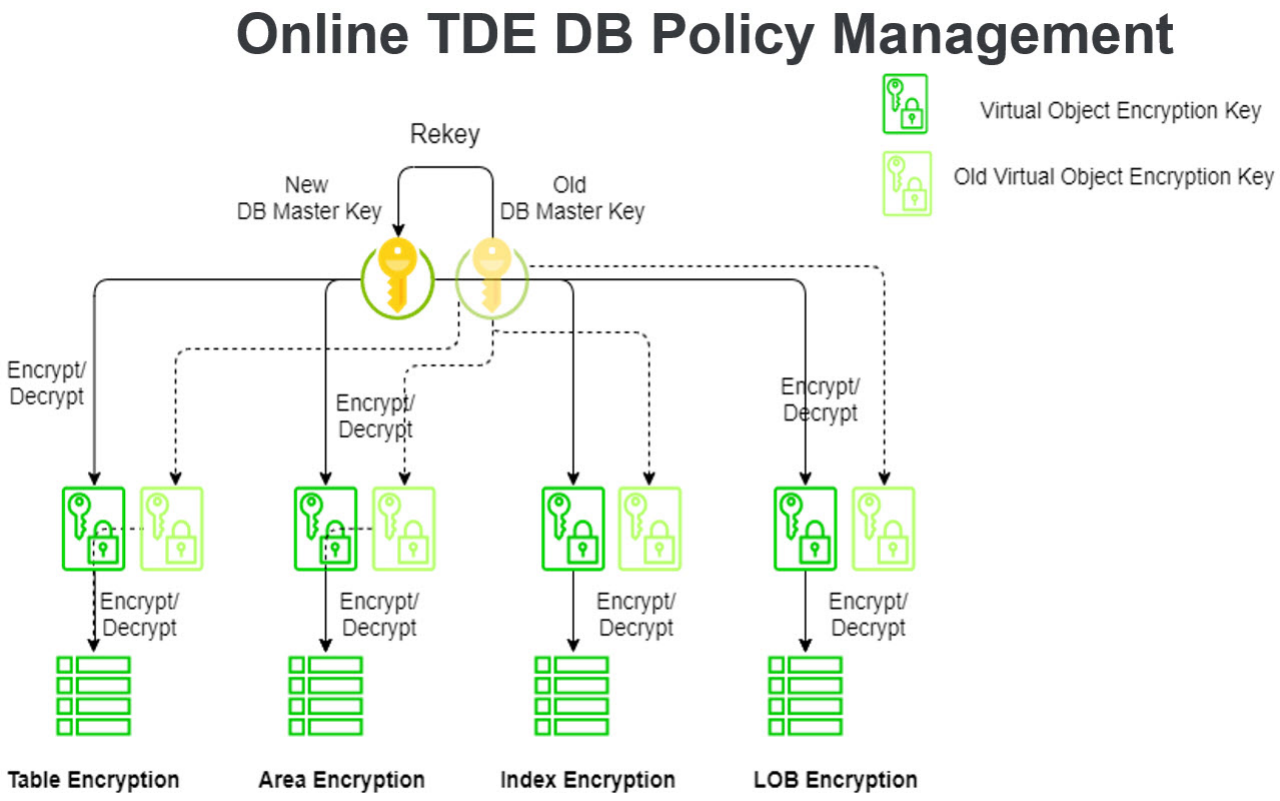
OpenEdge databases that use TDE (Transparent Data Encryption) have an encryption policy for the database master key (DMK) known as the encryption DB policy, for short. Now a database administrator or security administrator can perform TDE DB policy management tasks on TDE-enabled databases while the database is running.

Although you should change object policies regularly for data protection, you should change the encryption DB policy in the following circumstances:

- If you deploy your product with a TDE-enabled template database, change the encryption DB policy so that each installation has a unique set of object encryption keys.
- If your enterprise security policies need larger master keys, or keys with a different algorithm type, change the encryption DB policy.
- If you think that backup copies of both the database and the keystore have been compromised, change the encryption DB policy as insurance.

If you need to change the encryption DB policy for one of these reasons, you can also change the cipher for certain database objects at the same time.

The following figure shows how changing the database master key creates a unique set of object encryption keys for database object (tables, areas, indexes, and LOBs).



See a video on online TDE DB policy management here: https://www.youtube.com/watch?v=-XtoB9NYIvo

The demos in the video show two ways to change encryption DB policies:

- Create a new current encryption DB policy.
- Create a new pending encryption DB policy. The example shows how to update objects to use the new DB policy and how to retire the previous policy after objects are updated to use the new policy.

For more information, see 'Online TDE DB policy management' in *Manage the OpenEdge Database*.

Manage encryption DB policies

For details, see:

- 'Enable TDE DB policy management' in *Manage the OpenEdge Database*.
- 'Change the database master key' in *Manage the OpenEdge Database*.
- 'Disable TDE DB policy management' in *Manage the OpenEdge Database*.
- 'View and scan the encryption DB policy' in *Manage the OpenEdge Database*.
- 'Retire an encryption DB policy' in *Manage the OpenEdge Database*.

Troubleshooting

Troubleshooting tools and processes for databases that use TDE are the same as in previous releases. This secction lists recommendations specific to TDE DB policy management in this release of OpenEdge.

Issues typically fall into one of these areas:

- Failure to enable TDE DB policy management
- Failure to disable TDE DB policy management
- Pending encryption DB policy needs removal
- Previous encryption DB policy needs retirement
- Cipher selection for the encryption DB policy
- Creation and activation of a pending encryption DB policy

Use Case: Correct problems enabling TDE DB Policy management

To change the DMK for databases in OpenEdge 12.4 or later releases, you must first enable TDE DB policy management in the databases.To change the DMK for databases in OpenEdge 12.4 or later releases, you must first enable TDE DB policy management in the databases.

To check if TDE DB policy management is enabled, use this command:

```
proutil dbname -C dbdes

ID   Feature                           Active  Details
  ----  --------------------------------  ------  -------
5  Large Files                       Yes
     9  64 Bit DBKEYS                     Yes
    10  Large Keys                        Yes
    11  64 Bit Sequences                  Yes
    13  Encryption                        Yes
    29  TDE DB Policy Management          Yes
```

If TDE DB policy management is not enabled, enable it as follows:

```
proutil dbname -C enabletdedbpolicymanagement
```

You'll see the following message after enablement:

```
TDE DB Policy Management has been enabled for database testdb. (12479)
```

If enablement failed, check the error message.

Common failures are:

- `Keystore Admin Privileges are required for this operation`

  Check that passphrase used in the operation is the admin passphrase.

- `Lock table overflow, increase -L on server`

  Restart the database with larger -L parameter value and try again.

If the operation failed without an error message about the cause, restart the database to go through crash recovery and run idxcheck on the encryption policy tables `_sec-db-policy` and `_sec-obj-policy`. The `enabletdedbpolicymanagement` option updates the policy records to OpenEdge 12.4 format. If there are index or record corrections, they may cause the operation to fail.

- If idxcheck finds index errors, rebuild the indexes and retry `enabletdedbpolicymanagement`.
- Use dbtool to check for record corruptions. If it finds an error, back up encryption policies, then disable and re-enable encryption. After successfully re-enabling encryption, reapply the encryption policies. The TDE DB policy management feature is enabled along with encryption.

If these recommendations do not solve the issue, send the DB log file, keystore, and log of EPOLICY VIEW on the database and objects to Progress Technical Support for investigation.

Use Case: Correct problems disabling TDE DB policy management

Like the enablement, disablement of TDE DB policy management may fail if you don't provide the admin passphrase or if the value of the -L parameter is too small. Check the error message and use the correct admin passphrase or a larger -L setting, according to the error message. If the disablement failed or crashed without an error, check for corruption of encryption policy records and indexes.

The operation also updates the key store of the database. Prior to the update, it makes an OS copy of the key store file and saved to `dbname.ksbk.pid.date`, for example, `testdb.ksbk.6927.Wed_Jun_9_11_36_44_2021`.

If the disablement failed and the backup file was created, you must restore the keystore from the backup:

```
cp dbname.ksbk.pid.date dbname.ks
```

Use Case: Delete an unnecessary pending encryption DB policy

If you don't need the pending encryption DB policy for activation, there two ways to delete it:

- Run EPOLICY MANAGE DBPOLICY RETIRE:

  ```
  Proutil <dbname> -C epolicy manage dbpolicy retire
  ```

  This command deletes the pending encryption DB policy and its associated encryption object policies.

- Run EPOLICY MANAGE DBPOLICY REKEY:

  ```
  Proutil <dbname> -C epolicy manage <objtype> update <objname>
  ```

  If there are associated object policies with a previous encryption DB policy, use EPOLICY MANAGE UPDATE for each encryption object policy.

  ```
  Proutil <dbname> -C epolicy manage <objtype> update <objname>
  ```

  Once you run EPOLICY MANAGE UPDATE, there will be no object policy associated with the previous DB policy. Now use either method above to retire the previous encryption DB policy.

Use Case: Retire the previous encryption DB policy

If you need a new encryption DB policy, you must first set the previous encryption DB policy state to retired.

There are two ways to retire a previous encryption DB policy that has no associated object policies:

- Run EPOLICY MANAGE DBPOLICY RETIRE:

  ```
  Proutil <dbname> -C epolicy manage dbpolicy retire
  ```

  This command retires the previous encryption DB policy.

- Run EPOLICY MANAGE DBPOLICY REKEY:

  ```
  Proutil <dbname> -C epolicy manage dbpolicy rekey
  ```

  This command retires the previous DB policy. If you also want to create a new encryption DB policy, continue to run the utility once the previous encryption DB policy is retired.

If there are associated object policies with previous encryption DB policy, use EPOLICY MANAGE UPDATE for each encryption object policy.

```
Proutil <dbname> -C epolicy manage <objtype> update <objname>
```

Once you run EPOLICY MANAGE UPDATE, there is no object policy associated with the previous DB policy. You can now use either of the above methods to retire the previous encryption DB policy.

Use Case: Cipher selection for the encryption DB policy

If the cipher in use is compromised, create a new encryption DB policy with a new cipher as follows:

```
proutil db-name -C epolicy manage dbpolicy rekey [-Cipher <CipherName>] -Passphrase <
pass.txt>
```

The `-Cipher` parameter specifies the new cipher for the DB encryption policy. If you don't provide a `-Cipher` parameter, the utility uses the current DB policy's cipher, and returns this message:

```
Encryption DB Policy’s cipher is not specified, operation will use existing DB Policy’s
 cipher: <ciphername>. (20121)
Do you want to continue (y/n)?
```

Use Case: Creation and activation of a pending encryption DB policy

To create a pending encryption DB policy, and then activate it, enter the following command:

```
proutil db-name -C epolicy manage dbpolicy rekey pending [-Cipher <CipherName>] -Passphrase
 < pass.txt>
```

The `-Cipher` parameter specifies the cipher for the pending DB encryption policy. If the cipher is a PBE cipher, you won't need a passphrase when creating a pending encryption DB policy, but a passphrase is required to activate the pending DB encryption policy.

To activate the pending encryption DB policy, enter:

```
proutil db-name -C epolicy manage dbpolicy activate -Passphrase <pass.txt>
```

If you use OpenEdge Replication, see the topic *Troubleshoot TLS issues for OpenEdge Replication* in *Manage the OpenEdge database*.

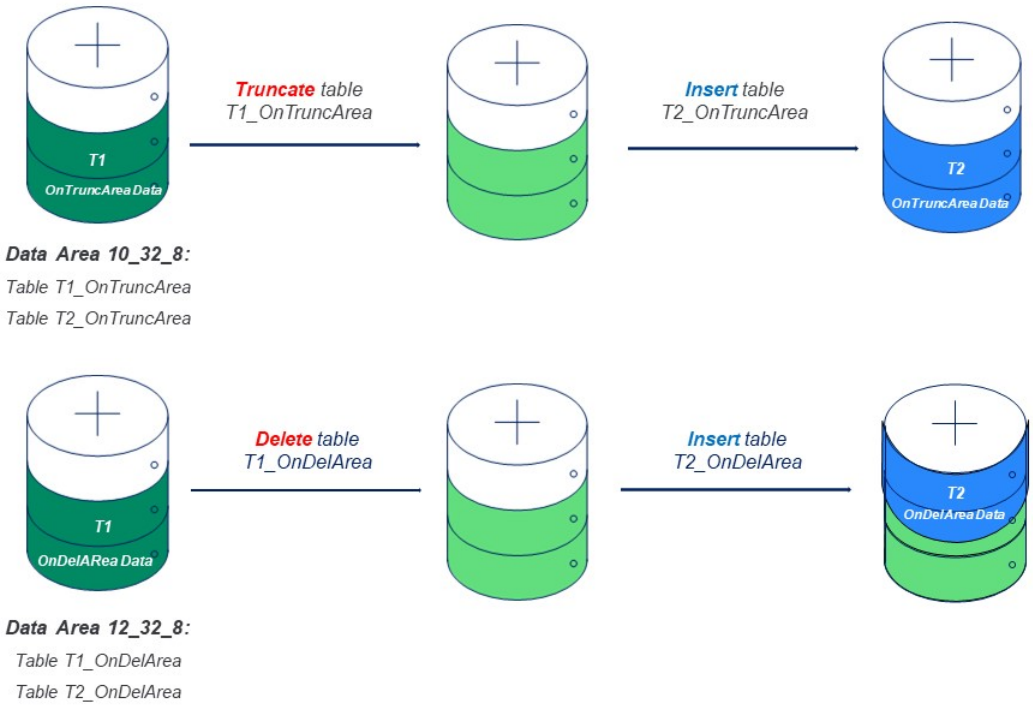
Online table truncation

For large history tables, or for cost savings in cloud deployments, a database administrator may need to truncate all the data associated with a particular table while the database is running. Now you can quickly and efficiently truncate all the data associated with a particular table to regain the space used.

Previous releases offered two ways to regain space:

- Remove data records and index entries one by one using a language program, which did not return storage space for reuse by other objects, or
- Use a language program to dump schema definitions and remove the table, then drop and recreate and reload the table schema. This is usually a slow offline process that requires rebasing replication targets.

In this release of OpenEdge, the PROUTIL utility allows you truncate, and optionally move, a table while the database is running, without having to recreate the schema.



See a video on Truncate table: https://www.youtube.com/watch?v=9oOZt5Xc_ZQ

How to truncate a table online

For more information, see 'Truncate tables' in *Manage the OpenEdge database*.

Troubleshooting

In general, PROUTIL TABLEMOVE TRUNCATE does not affect running applications, but it acquires an exclusive lock on the table, so it will introduce contention for the table's data being truncated. Requests for data from the ABL after truncation return a `stale cursor` error or return no data. No application triggers fire as the data is removed at the object level and not the record level. This may introduce referential integrity issues, so make certain that you understand the relationship on the table being truncated.

PROUTIL TABLEMOVE TRUNCATE is logged in the `.lg` file and by the KeyEvents facility.

Database maintenance mode

Database administrators can now perform maintenance tasks without worrying about incomplete transactions or new logins during administrative operations.

Tasks like purge, mass load, or synchronizing replication source to target are difficult when the database is still open for end users to log in. A database administrator may want to do a large maintenance task, but not expose it to end users until it's ready.

This release includes a restricted online mode, so the database administrator can run maintenance tasks with more of the buffer pool and helper processes available, improving the performance for these tasks. The tasks in restricted mode can also be run offline. When this mode is enabled, only database utilities can connect to the database.

See a video that introduces database maintenance mode here: https://www.youtube.com/watch?v=LJMJiUhue-c

See a video that demonstrates how to enable and use database maintenance mode here: https://www.youtube.com/watch?v=_zhY0sqQRXU

For details, see 'PROUTIL DBRESTRICT qualifier' in *Manage the OpenEdge Database*.

How to use restricted online mode

For details, see:

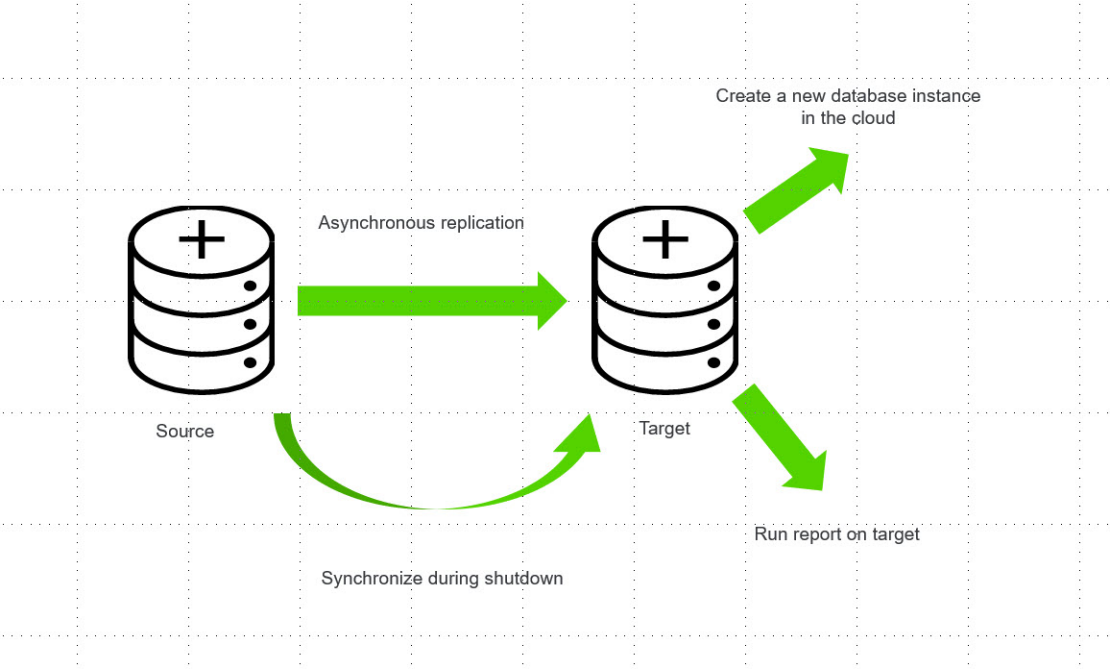
- 'Restrict database access' in *Manage the OpenEdge Database*
- 'View restricted mode status' in *Manage the OpenEdge Database*
- 'Change from one restricted mode to another' in *Manage the OpenEdge Database*

Troubleshooting

For details, see 'Troubleshoot restricted mode status' in *Manage the OpenEdge Database*.

Synchronize database replication source with target databases

A guaranteed synchronization state between the replication source and target databases lets a database administrator perform critical administrative actions on databases and can help expedite replication processes, such as transition, that require synchronization. For example, synchronization guarantees that read-only clients running on the replication target database will have consistent data with the replication source database.



See a video on synchronize database replication source with target databases here: https://www.youtube.com/watch?v=StWtd8jPq2c

For more information, see 'Enforce source and target database synchronization during shutdown' in *Use Database Replication*.

How to use enforced synchronization of replication source and target databases

For details, see:

- 'Enforce synchronization of replication source and target databases' in *Manage the OpenEdge Database*
- 'PROSHUT command' in *Manage the OpenEdge Database*

Troubleshooting

If your source database shutdown fails during synchronization, check your database log file for messages.

To check the success of synchronization, search the replication source database logs for messages like the following:

Error code	Message	Result
20258	Database broker has verified that all written AI locations were sent to replication target databases.	Synchronization succeeded.
10436	The source database [db-name] and the target database [db-name] on host [host-name] are synchronized.	Synchronization succeeded.
20257	Replication server has verified that all connected replication target databases are synchronized with the replication source database.	Synchronization succeeded.
20250	All connected replication target databases are synchronized with the replication source database. Normal shutdown will continue.	Synchronization succeeded.
20248	Replication synchronization during database shutdown [completed successfully | failed].	Synchronization succeeded.
20251	Replication server could not confirm as to whether connected replication target databases were in synchronization with source database.	Synchronization failed. Normal shutdown will continue.
20252	Replication server is not connected to any replication target databases.	Synchronization failed. Normal shutdown will continue.
10274	Invalid handle of replication server was received.	Synchronization failed. Normal shutdown will continue.
20255	No connected replication agents detected during shutdown synchronization.	Synchronization failed.

PAS for OpenEdge

ABL Service Deployment Packaging

An ABL service is a set of business logic that can be accessed through a URL path. The ABL Service Deployment feature allows users to package their ABL services into one deployment action, which allows users to build a custom service quickly and to easily deploy to a PAS for OpenEdge instance. This creates a single archive file (`.oeds`) for each transport, which makes the ABL Service easy to maintain and benefit from CI/CD processes.

See a video on ABL Service Deployment here: https://www.youtube.com/watch?v=PIdwh5Gi4YU

For more information, see 'ABL Service Deployment' in *Manage Progress Application Server for OpenEdge*.

How to use ABL Service Deployment

ABL Service Deployment creates a `.oeds` file that allows you to:

- Package an ABL service into one archive file
- Easily deploy your service to a PAS for OpenEdge instance
- Build a custom service quickly
- Easily maintain your service and benefit from CI/CD

For more information, see 'ABL Service Deployment' in *Manage Progress Application Server for OpenEdge*.

Troubleshooting

If there are issues with deployment, check your `manifest.mf` file for the following issues:

- The `ABL-Service-Name` value should match the folder name.
- The `ABL-Service-Version` should match the version number in the filename.

Enhanced troubleshooting for PAS for OpenEdge multi-session agent

Initiation of graceful emergency shutdown of multi-session agent was introduced in OpenEdge 12.3. In this release of OpenEdge, the feature is enhanced to include additional information by generating trace files. The trace files collect deferred logs, ABL call stacks of other running requests, and other information that enables you to debug and identify the root cause of the failure.

See a video on Deferred logging information in PAS for OpenEdge trace files here: https://www.youtube.com/watch?v=TFxXuSONHOM

How to enable deferred logging in PAS for OpenEdge

To enable deferred logging in PAS for OpenEdge to get more information about graceful emergency shutdown, follow:

- The Deferred Logging feature is configured in the *`CATALINA_BASE`*`/conf/openedge.properties` file.
- The following three properties control how much in-memory logging information is stored:

1. `defrdLogNumLines`
2. `defrdLoggingLevel`
3. `defrdLogEntryTypes`

- The deferred logging buffer is enabled only if all the three properties are set to a non-zero value.

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

Troubleshooting

For more troubleshooting information, see 'Graceful emergency shutdown of a multi-session agent' in *Manage Progress Application Server (PAS) for OpenEdge*.

OpenEdge Command Center

Overview of the OpenEdge Command Center

The OpenEdge Command Center is a cloud-ready OpenEdge management console capable of managing multiple OpenEdge resources and installations across various machines and versions of OpenEdge. OpenEdge Command Center features the latest UI and UX technologies. In its first release, OpenEdge Command Center is built to effectively manage PAS for OpenEdge instances and, in the future, databases. It aims to incrementally support everything you need to manage instances. 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 allowing for a secure connection 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 with interactive swagger documentation to help you add your own resources along the way.

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. Built for high availability, the OpenEdge Command Center server can be deployed on multiple servers, managed by a load balancer, and unified through a common database. Built-in user management for system administrators and application administrators ensures that you have secure control of OpenEdge Command Center user permissions across all of your instances. For complete resource visibility, OpenEdge Command Center offers automated component discovery for OpenEdge version 12.2.5 and later releases for all PAS for OpenEdge deployments.

Get started with the OpenEdge Command Center

For more information, see https://www.youtube.com/watch?v=1DxBF1U35vk

For more information, see the topic 'Learn about OpenEdge Command Center' in *Use the OpenEdge Command Center*.

Troubleshooting

Additional troubleshooting content to be posted on or before July 8th, as available.

Install OpenEdge Command Center on a Linux server

For more information, see https://www.youtube.com/watch?v=rd-roeEVCMc

For more information, see the topic *Install and configure OpenEdge Command Center* in *Use the OpenEdge Command Center*.

Troubleshooting

Additional troubleshooting content to be posted on or before July 8th, as available.

Configure MongoDB

For more information, see https://www.youtube.com/watch?v=BUTtQwxLqNc

For more information, see the topic *Set up OpenEdge Command Center with MongoDB Atlas on AWS* in *Use the OpenEdge Command Center*.

Troubleshooting

Additional troubleshooting content to be posted on or before July 8th, as available.

Manage PAS for OpenEdge instances

For more information, see https://www.youtube.com/watch?v=gqJkHiizHHg

For more information, see the topic *Configure new OpenEdge Command Center agents* in *Use the OpenEdge Command Center*.

Troubleshooting

Additional troubleshooting content to be posted on or before July 8th, as available.

User management

For more information, see https://www.youtube.com/watch?v=jFug5b-hp3k

For more information, see the topic *Users* in *Use the OpenEdge Command Center*.

Troubleshooting

Additional troubleshooting content to be posted on or before July 8th, as available.

Super admin password utility

For more information, see https://www.youtube.com/watch?v=Wfy-MvVmVKY

For more information, see *Use the OpenEdge Command Center*.

Troubleshooting

Additional troubleshooting content to be posted on or before July 8th, as available.