Progress®



Optimize the OpenEdge Database for Continuous Operation

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

Updated: 2021/06/28

Table of Contents

How OpenEdge databases support continuous operation.....9

Change database parameters while the database is running.....11

Modify database startup parameters online using PROMON.....19

PROUTIL INCREASETO qualifier.....21

Change schemas while the database is running.....25

ABL online schema changes to triggers.....26

Change non-structural fields of _File and _Field online.....26

Change Int to Int64 online.....28

Change default storage areas for created objects while the database is running.....29

Truncate and move tables while the database is running.....30

Optimize database client notifications.....31

Manage brokers for continuous operation.....33

Enable automatic database reconnection.....35

Manage OpenEdge replication for continuous operation.....41

OpenEdge Replication DSRUTIL Properties Validation Utility.....42

_Repl-Server VST field descriptions.....43

Database status (_DbStatus).....50

Update ABL client applications.....53

Change ABL triggers online.....53

Rename fields using ABL.....55

Drop a field online.....56

Preface

Purpose

This guide describes OpenEdge® RDBMS features that support continuous operation. The procedures described help you to minimize database down time and disruption. This manual assumes that you are familiar with the OpenEdge RDBMS planning concepts discussed in *OpenEdge Getting Started: Database Essentials*.

Audience

This book is intended for OpenEdge database administrators and OpenEdge application developers.

For the latest documentation updates go to https://docs.progress.com.

Organization

How OpenEdge databases support continuous operation on page 9

Provides an overview of built-in features that enable continuous operation.

Change database parameters while the database is running on page 11

Lists parameters that you can change while the database runs and explains how to change them.

Change schemas while the database is running on page 25

Explains how to change triggers, non-structural fields, and certain datatypes that do not require schema locks.

Optimize database client notifications on page 31

Describes how to use startup parameters, the PROMON utility, and VSTs to control the method and frequency of notification delivery, which reduces network traffic and improves response time.

Manage brokers for continuous operation on page 33

Describes how to troubleshoot broker activity in the database log, keep secondary brokers running during recovery transitions, and disconnect secondary brokers without a process kill signal.

Enable automatic database reconnection on page 35

Explains how to set up a connection set, so that, if the connection to the primary database fails, client-server connections can automatically connect to an alternate database, such as a replication target or backup database.

Manage OpenEdge replication for continuous operation on page 41

Describes how to use validate the DSRUTIL utility to find replication properties of replication-enabled source and target databases and confirm running status of target database brokers.

Update ABL client applications on page 53

Explains how to change trigger definitions or rename user table fields without requiring exclusive schema locks, and how to rename fields without disconnecting users from the database.

Documentation conventions

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

1

How OpenEdge databases support continuous operation

To meet the ever-growing demand for 'five nines' of service, the OpenEdge database is designed to ensure continuous operation.

This document provides an overview of built-in features that you can leverage to make sure that applications and services depending on the OpenEdge database exceed user expectations for responsiveness and reliability.

For more information, see:

- Change database parameters while the database is running on page 11
- Change schemas while the database is running on page 25
- Optimize database client notifications on page 31
- Perform maintenance while the database is running
- Manage brokers for continuous operation on page 33
- Enable automatic database reconnection on page 35
- Manage encryption DB policies while the database is running
- Manage OpenEdge replication for continuous operation on page 41
- Update ABL client applications on page 53

Learn how to minimize disruption by optimizing client application and schema updates, tuning database parameters while the database is running, enabling automatic database reconnection, and managing replication failover more efficiently.

2

Change database parameters while the database is running

You can change most parameters using either PROMON or VSTs, but those that require resources are changed using the PROUTIL INCREASETO command.

The following table lists database parameters that you can modify while the database is running and tells how to change them. Links to details about the change procedures follow the table.

Table 1: Database server continuous operation parameters

Parameter	Update Method	Notes
After-image File Management Archive Directory List (-aiarcdir)	PROMON or `_DbParams`	You can only change this parameter for the primary broker.
After-image File Management Archive Directory Create (-aiarcdircreate)	PROMON or `_DbParams`	You can only change this parameter for the primary broker.
After-image File Management Archive Interval (-aiarcinterval)	PROMON or `_DbParams`	You can only change this parameter for the primary broker.
After-image Buffers (-aibufs)	PROUTIL INCREASETO	You can only change this parameter for the primary broker.
After-image Stall (-aistall)	PROMON or `_DbParams`	You can only change this parameter for the primary broker.
Blocks in Database Buffers -B)	PROUTIL INCREASETO	You can only change this parameter for the primary broker.

Parameter	Update Method	Notes
Blocks in Alternate Buffer Pool (-B2)	PROUTIL INCREASETO	You can only change this parameter for the primary broker.
Base Index (-baseindex)	PROMON or _dbParams	You can only change this parameter for the primary broker.
Base LOB (-baselob)	PROMON or _dbParams	You can only change this parameter for the primary broker.
Base Table (-basetable)	PROMON or _dbParams	You can only change this parameter for the primary broker.
Base user index (-baseuserindex)	PROMON or _dbParams	You can only change this parameter for the primary broker.
Base user LOB (-baseuserlob)	PROMON or _dbParams	You can only change this parameter for the primary broker.
Base user table (-baseusertable)	PROMON or _dbParams	You can only change this parameter for the primary broker.
Before-image Buffers (-bibufs)	PROUTIL INCREASETO	You can only change this parameter for the primary broker.
Before-image scan time (-biscantime)	PROMON or _dbParams	You can only change this parameter for the primary broker.
Threshold Stall (-bistall)	PROMON or _dbParams	You can only change this parameter for the primary broker.
Recovery Log Threshold (-bithold)	PROMON or _dbParams	You can only change this parameter for the primary broker.
Maximum Private Buffers per Connection (-Bpmax)	PROMON or _dbParams	You can only change this parameter for the primary broker.
CDC cache size (-cdcsize)	PROUTIL INCREASETO	You can only change this parameter for the primary broker.
C stack print option (-cstackPrintopt)	PROMON or _dbParams	You can only change this parameter for the primary broker.
Database consistency check (-DbCheck)	PROMON or _dbParams	You can only change this parameter for the primary broker.
Database Notification Operations (-dbnotifyops)	PROMON or _dbParams	You can only change this parameter for the primary broker.
Database Notification Time (-dbnotifytime)	PROMON or _dbParams	You can only change this parameter for the primary broker.

Parameter	Update Method	Notes
Diagnostic directory (-diagDir)	PROMON or _dbParams	You can only change this parameter for the primary broker.
Diagnostic events value (-diagEvent)	PROMON or _dbParams	You can only change this parameter for the primary broker.
Diagnostic event level (-diagEvtLevel)	PROMON or _dbParams	You can only change this parameter for the primary broker.
Diagnostic data format (-diagFormat)	PROMON or _dbParams	You can only change this parameter for the primary broker.
Diagnostic field separator (-diagFS)	PROMON or _dbParams	You can only change this parameter for the primary broker.
Diagnostic pause length (-diagPause)	PROMON or _dbParams	You can only change this parameter for the primary broker.
Diagnostic prefix value (-diagPrefix)	PROMON or _dbParams	You can only change this parameter for the primary broker.
Encryption cache size (-ecsize)	PROUTIL INCREASETO	You can only change this parameter for the primary broker.
Group delay (-groupdelay)	PROMON or _dbParams	You can only change this parameter for the primary broker.
Index consistency check (-IndexCheck)	PROMON or _dbParams	You can only change this parameter for the primary broker.
Index Range Size (-indexrangesize)	PROUTIL INCREASETO	You can only change this parameter for the primary broker.
Lock Table Entries (-L)	PROUTIL INCREASETO	You can only change this parameter for the primary broker.
Lock Governor (-LGovernor)	PROMON or _dbParams	You can only change this parameter for the primary broker.
Log file archive directory (-lgArchiveDir)	PROMON or _dbParams	You can only change this parameter for the primary broker.
Log file archive enable (-lgArchiveEnable)	PROMON or _dbParams	You can only change this parameter for the primary broker.
Log file truncate frequency (-lgTruncateFrequency)	PROMON or _dbParams	You can only change this parameter for the primary broker.
Log file truncate size (-lgTruncateSize)	PROMON or _dbParams	You can only change this parameter for the primary broker.

Parameter	Update Method	Notes
Log file truncate time (-lgTruncateTime)	PROMON or `_dbParams`	You can only change this parameter for the primary broker.
Limit log file payload (-limitLgPayload)	PROMON or `_dbParams`	You can only change this parameter for the primary broker.
LOB range (-lobrangesize)	PROUTIL INCREASETO	You can only change this parameter for the primary broker.
LRU force skips (-lruskips)	PROMON or `_dbParams`	You can only change this parameter for the primary broker.
Maximum Clients per Server (-Ma)	PROMON or `_dbParams` for primary broker, PROMON or `_Servers` for secondary broker.	You can change this parameter for the primary or secondary broker.
Maximum Dynamic Server (-maxport)	PROMON or `_dbParams` for primary broker, PROMON or `_Servers` for secondary broker.	You can change this parameter for the primary or secondary broker.
Memory consistency check (-Memcheck)	PROMON or `_dbParams`	You can only change this parameter for the primary broker.
Delayed BI File Write (-Mf)	PROMON or `_dbParams`	You can only change this parameter for the primary broker.
Minimum Clients per Server (-Mi)	PROMON or `_dbParams` for primary broker, PROMON or `_Servers` for secondary broker.	You can change this parameter for the primary or secondary broker.
Minimum Dynamic Server (-minport)	PROMON or `_dbParams` for primary broker, PROMON or `_Servers` for secondary broker.	You can change this parameter for the primary or secondary broker.
Message Buffer Size (-Mm)	PROMON or `_dbParams` for primary broker, PROMON or `_Servers` for secondary broker.	You can change this parameter for the primary or secondary broker.
Maximum Servers (-Mn)	PROUTIL INCREASETO	You can only change this parameter for the primary broker.
Maximum Servers per Broker (-Mpb)	PROMON or `_dbParams` for primary broker, PROMON or `_Servers` for secondary broker.	You can change this parameter for the primary or secondary broker.
Multi-tenancy Partition Cache Size (-mtpmsize)	PROUTIL INCREASETO	You can only change this parameter for the primary broker.
Shared-memory Overflow Size (-Mxs)	PROUTIL INCREASETO	You can only change this parameter for the primary broker.

Parameter	Update Method	Notes
Number of Users (-n)	PROUTIL INCREASETO	You can only change this parameter for the primary broker.
Nap Time (-nap)	PROMON or `_dbParams`	You can only change this parameter for the primary broker.
Nap Maximum (-napmax)	PROMON or `_dbParams`	You can only change this parameter for the primary broker.
Nap Time Steps (-napstep)	PROMON or `_dbParams`	You can only change this parameter for the primary broker.
Login Governor (-nGovernor)	PROMON or `_dbParams`	You can only change this parameter for the primary broker.
Message Wait (-Nmsgwait)	PROMON or `_dbParams`	You can only change this parameter for the primary broker.
No Host Verify (-nohostverify)	PROMON or `_dbParams`	You can only change this parameter for the primary broker.
No Session Cache (-nosessioncache)	PROMON or `_dbParams`	You can only change this parameter for the primary broker.
No Session Reuse (-nosessionreuse)	PROMON or `_dbParams`	You can only change this parameter for the primary broker.
Number of checkpoint statistics (-numCheckpointStats)	PROMON or `_dbParams`	You can only change this parameter for the primary broker.
Omit Log Messages (-omitLgMsgs)	PROMON or `_dbParams`	You can only change this parameter for the primary broker.
Storage Object Cache Size (-omsize)	PROUTIL INCREASETO	You can only change this parameter for the primary broker.
Pending Connection Time (-PendConnTime)	PROMON or `_dbParams` for primary broker, PROMON or `_Servers` for secondary broker.	You can change this parameter for the primary or secondary broker.
Database Service Communication Area Size (-pica)	PROUTIL INCREASETO	You can only change this parameter for the primary broker.
Database PICA Queue Nap Time (-picanap)	PROMON or `_dbParams`	You can only change this parameter for the primary broker.
Pin Shared Memory (-pinshm)	PROMON or `_dbParams`	You can only change this parameter for the primary broker.
Prefetch delay (-prefetchDelay)	PROMON or `_dbParams`	You can only change this parameter for the primary broker.

Parameter	Update Method	Notes
Prefetch Factor (-prefetchFactor)	PROMON or _dbParams	You can only change this parameter for the primary broker.
Prefetch Num Recs (-prefetchNumRecs)	PROMON or _dbParams	You can only change this parameter for the primary broker.
Prefetch Priority (-preFetchPriority)	PROMON or _dbParams	You can only change this parameter for the primary broker.
Record free chain search depth factor (-recspacesearchdepth)	PROMON or _dbParams	You can only change this parameter for the primary broker.
Schema Lock Wait Queue (-schlockwq)	PROMON or _dbParams	You can only change this parameter for the primary broker.
Security cache size (-secsize)	PROUTIL INCREASETO	You can only change this parameter for the primary broker.
Semaphore Sets (-semsets)	PROMON or _dbParams	You can only change this parameter for the primary broker.
Type-of-Server-to-Start-ServerType.html	PROMON or _dbParams	You can only change this parameter for the primary broker.
Shared memory segment size (-shmsegsize)	PROMON or _dbParams	You can only change this parameter for the primary broker.
Spin Lock Retries (-spin)	PROMON or _dbParams	You can only change this parameter for the primary broker.
Autonomous Update Statistics (-SQLAutoStats)	PROMON or _dbParams	You can only change this parameter for the primary broker.
SQL Open Cursors (-SQLCursors)	PROMON or _dbParams	You can only change this parameter for the primary broker.
Lock Wait Timeout(-SQLLockWaitTimeout)	PROMON or _dbParams	You can only change this parameter for the primary broker.
SQL rowid identifier (-SQLQuotedRowid)	PROMON or _dbParams	You can only change this parameter for the primary broker.
SQL Stack Size (-SQLStack)	PROMON or _dbParams	You can only change this parameter for the primary broker.
SQL Statement Cache Size (-SQLStmtCache)	PROMON or _dbParams	You can only change this parameter for the primary broker.
SQL Sorting Memory (-SQLTempBuff)	PROMON or _dbParams	You can only change this parameter for the primary broker.

Parameter	Update Method	Notes
SQL Sorting on Disk (-SQLTempDisk)	PROMON or `_dbParams`	You can only change this parameter for the primary broker.
SQL Temp Table Data Page (-SQLTempPageSize)	PROMON or `_dbParams`	You can only change this parameter for the primary broker.
Authorized Data Truncation (-SQLTruncateTooLarge)	PROMON or `_dbparams`	You can only change this parameter for the primary broker.
Autonomus Schema Update (-SQLWidthUpdate)	PROMON or `_dbParams`	You can only change this parameter for the primary broker.
Server-side join (-ssj)	PROMON or `_dbParams`	You can only change this parameter for the primary broker.
TLS Encryption for TCP/IP Connections (-ssl)	PROMON or `_dbParams`	You can only change the `-ssl` parameter for new servers. the broker fins an appropriate server for mixed `-ssl` setting.
STS logging level (-stsloggінglevel)	PROMON or `_dbParams`	You can only change this parameter for the primary broker.
Table consistency check (-TableCheck)	PROMON or `_dbParams`	You can only change this parameter for the primary broker.
Table Range Size (-tablerangesize)	PROUTIL INCREASETO	You can only change this parameter for the primary broker.
Transaction End (TXE) Lock Retry Limit (-TXERetryLimit)	PROMON or `_dbParams`	You can only change this parameter for the primary broker.
User index range (-userindexrangesize)	PROUTIL INCREASETO	You can only change this parameter for the primary broker.
User LOB range (-userlobrangesize)	PROUTIL INCREASETO	You can only change this parameter for the primary broker.
Database Client Notification (-usernotifytime)	PROMON or `_dbParams`	You can only change this parameter for the primary broker.
User Table Range (-usertablerangesize)	PROUTIL INCREASETO	You can only change this parameter for the primary broker.

To change parameters using PROMON, see 'Modify database startup parameters online using PROMON' in *Optimize the OpenEdge Database for Continuous Operation*.

To change parameters using virtual system tables, use the appropriate VST for the desired broker:

- For primary broker parameters, use the _Dbparams VST. For example, you can use ABL code like this to set the value of the LRU force skips parameter (-lruskips) to 100 :

```
Find _dbParams
   where _dbParams-Name = "-lruskips".
   assign _dbParams-Value = 100.
```

  For details about _Dbparams, see 'Database startup parameters file status (_DbParams)' in *Manage the OpenEdge Database*.

- For the secondary broker parameters, use the _Servers VST. You need the index number for each parameter you want to change. Use a command like this to retrieve the index numbers:

```
DEF VAR i AS INT.
FOR EACH _Servers  WHERE _Server-Pid <> 0:
DISPLAY
  _Server-type FORMAT "x(8)" LABEL "Type".
  REPEAT i = 1 TO 64:
    IF (_SrvParam-Name[i] = ?) THEN
     LEAVE.
    ELSE
     DISPLAY
     i LABEL "Index" //This is the index of SrvParam-Value & _SrvParam-Name
    _SrvParam-Name[i] format "x(22)" label "Name"
    _SrvParam-Value[i] format "x(8)" label "Value"
    _SrvParam-dtype[i]  label "DataType"
    _SrvParam-Is-Modifiable[i] label "Mod"
    _SrvParam-Is-Default[i] label "Def".
  END.
END.
```

```
     Index Name                   Value    DataType Mod Def
---------- ---------------------- -------- -------- --- ---
         1 -ipver                 IPV4            1 no  no
         2 -maxport               2000            4 yes yes
         3 -minport               1025            4 yes yes
         4 -ssj                   1               3 yes yes
         5 -threadedServer        1               3 no  yes
         6 -threadedServerStack   512             4 no  yes
         7 -Ma                    1               4 yes no
         8 -Mi                    1               4 yes yes
         9 -Mm                    8192            4 yes yes
        10 -Mp                    0               4 no  yes
        11 -Mpb                   5               4 yes no
        12 -N                     TCP             1 no  yes
        13 -PendConnTime          0               4 yes yes
        14 -S                     22202           1 no  no
        15 -ServerType            BOTH            4 no  no
        16 -SQLCursors            50              4 no  yes
        17 -SQLLockWaitTimeout    5               4 no  yes
        18 -SQLQuotedRowid        1               1 no  no
        19 -SQLStack              1000            4 no  yes
        20 -SQLStmtCache          100             4 no  yes
        21 -SQLTempBuff           1000            4 no  yes
        22 -SQLTempDisk           500000          4 no  yes
        23 -SQLTempPageSize       8               4 no  yes
        24 -SQLTruncateTooLarge   OFF             1 no  yes
        25 -SQLWidthUpdate        OFF             1 no  yes
```

  To update a parameter, use this syntax:

```
Find _Servers where _Server-Id = serverId.
_SrvParam-Value[serverParamIndex] = serverParamValue.
```

For example, to change the value of Maximum Servers per Broker (-Mpb) to 20, use the following, because the index value for -Mpb is 11:

```
Find _Servers where _Server-Id = 2.
_SrvParam-Value[11] = 20.
```

In the .lg file, a confirmation message appears:

```
(5647) Maximum Servers Per Broker (-Mpb): 20
```

For more details about _Servers, see 'Servers (_Servers)' in *Manage the OpenEdge Database*.

For PROUTIL syntax and details, see 'PROUTIL INCREASETO qualifier' in *Optimize the OpenEdge Database for Continuous Operation*.

To enable continuous operations, you can modify many database parameters while your database is running.

For details, see the following topics:

- Modify database startup parameters online using PROMON

Modify database startup parameters online using PROMON

This option enables you to modify specific database startup parameters while the database is online.

PROMON Adjust Startup Parameters display

The following shows a sample display of the PROMON utility menu when 16 is invoked from the **R&D Advanced Options** menu followed by 4 from the **Administrative Functions** menu.

```
OpenEdge Release 12 Monitor (R&D)
Adjust Startup Parameters

Create After-image Management Archival Directory(s) (-aiarcdircreate): Not Enabled
After-image Management Archival Interval (-aiarcinterval): -1
After-Image Stall (-aistall): Not Enabled
Starting index number for statistics range (-baseindex): 1
Starting LOB number for statistics range (-baselob): 1
Starting table number for statistics range (-basetable): 1
Starting index number per user for statistics range (-baseuserindex): 1
Starting LOB number per user for statistics range (-baselob): 1
Starting table number per user for statistics range (-baseusertable): 1
BI File Threshold Stall (-bistall): Disabled.
BI File Threshold size (-bithold): 0.0   Bytes
BIW writer delay (-bwdelay): 0
Diagnostic directory (-diagDir): Not Enabled
Diagnostic events value (-diagEvent): LockTable:0,BiThold:0,SysErr:0
Diagnostic event level (-diagEvtLevel): 0
Diagnostic field separator (-diagFS): ' '
Diagnostic data format (-diagFormat): csv
Diagnostic pause length (-diagPause): 0
Diagnostic prefix value (-diagPrefix): diagEvent_
Diagnostic procedure (-diagProc): Not Enabled

Enter a parameter name, <return>, R, P, T, or X (? for help):
```

Modify Database Parameters

Perform the following steps to modify a database startup parameter while the database is online:

1. Enter the parameter keyword including the dash (-) as in the example below.

```
Enter a parameter name, <return>, R, P, T, or X (? for help): -aistall
```

2. An error message is returned if the parameter is not valid.

```
Invalid parameter name. Please try again.
Enter a parameter name, <return>, R, P, T, or X (? for help):
```

Else if valid, PROMON will return the current value of the parameter in shared memory. Enter a new value for the parameter in the prompt as in the example below.

```
Current After-Image Stall (-aistall): Not Enabled
Enter a new value for -aistall (0 to 1): 1
```

3. If valid, PROMON returns a confirmation of the update to show the new value as in the example below.

```
Enter a new value for -aistall (0 to 1): 1
New After-Image Stall (-aistall): Enabled
```

Notes

- Only specific startup parameters may be modified while the database is online. See Change database parameters while the database is running on page 11 for more information on a specific parameter.
- PROMON lists those database startup parameters that are modified online. The full range is not displayed on one screen. You may scroll through the list of parameters and current values using <return> or else adjust the display line per PROMON menu to display all the parameters on one screen.
- Use R to return to the first page of parameters while scrolling.
- For database startup parameters that possess a range of values between a minimum and maximum, PROMON displays this range when prompting the user to enter a new value. For more information on these values, see Change database parameters while the database is running on page 11.
- When entering the name of a parameter to be changed, the parameter name must match exactly and include the dash (-), or PROMON returns an error.

- Any valid parameter name may be entered from any of the **Adjust Startup Parameters** menu pages. There is no need to be on the current page of the menu when entering a parameter name.
- Use (**CTRL+C**) at the prompt to cancel the operation.

PROUTIL INCREASETO qualifier

Increase certain server startup parameters while the database remains online.

Syntax

```
proutil db-name -C increaseto [ -n n ] [ -B n ][ -B2 n ] [ -L n ]
                [ -aibufs n ] [ -bibufs n ] [ -Mxs n ] [ -omsize n ] [
-mtpmsize n ]
                [ -cdcsize n ] [ -ecsize n ] [ -secsize n ] [
-indexrangesize n ]
                [ -userindexrangesize n ] [ -tablerangesize n ] [
-usertablerangesize n ]
                [ -pica n ]
```

Parameters

db-name

Specifies the name of the database to update.

`-n`

Use `-n` to increase the number of users.

`-B`

Use `-B` to specify the new value for the number of blocks in the primary buffer pool.

`-B2`

Use `-B2` to specify the new value for the number of blocks in the Alternate Buffer Pool.

`-L`

Use `-L` to specify the new limit of the record locking table.

`-aibufs`

Use `-aibufs` to specify the new number of after-image buffers.

`-bibufs`

Use `-bibufs` to specify the new number of before-image buffers.

`-Mxs`

Use `-Mxs` to specify the size of the shared-memory overflow in kilobytes.

`-omsize`

Use `-omsize` to specify the number entries in the secondary storage object cache. At startup, a primary and secondary storage object cache are allocated, each with the number of entries specified by the `-omsize` startup parameter; INCREASETO only modifies the size of the secondary cache.

`-mtpmsize`

Use `-mtpmsize` to specify the number of entries in the multi-tenancy partition cache. At startup, a primary and secondary multi-tenancy partition cache are allocated, each with the number of entries specified by the `-mtpmsize` startup parameter; INCREASETO only modifies the size of the secondary cache.

`-cdcsize`

Use `-cdcsize` to increase the size of the CDC Secondary cache.

`-ecsize`

Use `-ecsize` to increase the size of the encryption cache.

`-secsize`

Use `-secsize` to increase the size of the security cache.

`-indexrangesize`

Use `-indexrangesize` to increase the number of indexes for which you want to collect statistics.

`-userindexrangesize`

Use `-userindexrangesize` to increase the number of rows maintained in the`_UserIndexStat` VST.

`-tablerangesize`

Use `-tablerangesize` to increase the number of tables for which you want to collect statistics.

`-usertablerangesize`

Use `-usertablerangesize` to increase the number of rows maintained in the`_UserTableStat` VST.

`-pica`

Use `-pica` to increase the size of the database service communications area.

INCREASETO enables the identified startup parameters to be increased while the database is online. This improves database resiliency by minimizing planned downtime exclusively for changes to startup parameters, providing a robust, high availability mechanism that allows you to modify commonly used database resources while the database is online.

Notes

- You can execute any number of parameters in a single statement. For example:

```
proutil db-name -C increaseto -B 10240 -L 20480 -bibufs 360 -aibufs 500
```

- If a value specified to increase to is less than the current setting, an error is reported and the value is not changed.
- Only one instance of INCREASETO can connect to an online database at a time.
- Parameters specified using INCREASETO are restricted by currently supported limits. For example, the maximum increase for the number of blocks in the database buffer is limited to 1,000,000,000 for 64-bit platforms. See 'Database Startup Parameters' in *Manage the OpenEdge Database* for more information on range and limits of `-B`, `-L`, `-bibufs`, `-aibufs`, and `-Mxs`.
- You must have security privileges on the directly connected database to use INCREASETO.
- All users directly connected to the database must have privileges to read any new shared-memory segments created by INCREASETO. If a user does not have access, INCREASETO prompts you to disconnect the under-privileged users or abort the INCREASETO.

3

Change schemas while the database is running

You can use the SCHEMA-CHANGE attribute during an ABL session to perform a growing range of schema maintenance while the database is running.

The SCHEMA-CHANGE attribute relaxes the requirement for exclusive access to a database in order to make the following schema changes online:

- Add new sequences.
- Add new tables, as well as any associated fields, indexes, and database triggers (which must be added within the same transaction).
- Add new fields to an existing table.

  Note: When an add field operation is performed online, the connected clients can access (perform read/write operation on) the newly created field without disconnecting or refreshing their schema cache. For this to happen, during an online field creation, a DB notification is sent to all other connected clients, so that, they can refresh their schema cache at the end of their active transactions.

- Add new inactive indexes to an existing table.

The attribute must be set to `"NEW OBJECTS"` to activate this feature.

Notes

- For more information on the SCHEMA-CHANGE attribute, see 'SCHEMA-CHANGE attribute' in *ABL Reference*.

- For a list of schema change commands that can be executed against an online database using SQL, see 'Online schema changes' in *OpenEdge Data Management: SQL Development*.
- For more information on activating an inactive index, see 'Activate a single index' and 'PROUTIL IDXACTIVATE qualifier' in *Manage the OpenEdge Database*.

For details, see the following topics:

- ABL online schema changes to triggers
- Change non-structural fields of _File and _Field online
- Change Int to Int64 online
- Change default storage areas for created objects while the database is running
- Truncate and move tables while the database is running

ABL online schema changes to triggers

Online schema changes for ABL triggers enables you to change trigger definitions online without acquiring 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. Benefits of the feature include:

- Making changes to your ABL trigger schema is online by default, which allows you to make online schema changes for trigger operations even when other clients are connected to the same database
- You can also make changes to triggers using SQL
- You can make online trigger changes using the same tools and procedures used to perform offline trigger changes
- There is no need to re-compile your `r-code` to enable the use of the changed trigger definitions for ABL procedures
- Improved trigger responsiveness and availability in busy production environments

When the schema changer transaction begins the online schema change, it first checks the database to confirm that all other connected clients use OpenEdge 12.2 or higher. If any connected client is at a version less than 12.2, it cannot proceed with the schema change online, but must instead obtain an exclusive lock on the entire schema. During the short period of time that the schema change transaction is in process, new database logins and schema readers are blocked until the transaction completes.

Change non-structural fields of _File and _Field online

The non-structural fields of the `_File` and `_Field` tables may now be changed while the database remains running. Such changes no longer require an exclusive schema lock.

Note: During the short period of time that the schema change transaction is in process, new database logins and schema readers will be blocked until the transaction completes.

The following table lists the non-structural fields that you can change online without disrupting user experience:

Table 2: Non-structural fields that can be changed online

Schema Table	Field/Column Name
_Field	_Field._Col-label-SA
_Field	_Field._Col-label
_Field	* _Field._Fld-case
_Field	_Field._Format-SA
_Field	_Field._Format
_Field	_Field._Help-SA
_Field	_Field._Help
_Field	_Field._Label-SA
_Field	_Field._Label
_Field	* _Field._Valexp
_Field	_Field._Valmsg-SA
_Field	_Field._Valmsg
_Field	_Field._View-As _Field._Width
_File	_File._File-Label-SA
_File	_File._File-Label
_File	*_File._Valexp
_File	_File._Valmsg-SA
_File	_File._Valmsg

Effect on ABL applications

Unless users recompile their r-code, they will not see the effect of changes to non-structural fields.

For example, if an r-code is compiled before the column label is changed, then the old column label continues to appear until the r-code is recompiled. This does not affect how the business logic executes.

A recompilation of the r-code might be necessary for any changes to fields marked with an asterisk (*). If an affected field is referenced in the r-code, the r-code may execute differently. In this instance, any ABL application logic (r-code) that references these changed fields must be recompiled.

For example, if you modify the Valexp field in the _Field schema table, the AVM continues to execute the older validation expression until the r-code that accesses the database is recompiled.

Effect on transactional concurrency

The support for online changes to non-structural fields has the following effects on the concurrency of database transactions:

- Changing non-structural fields in the `_File` and `_Field` tables do not require exclusive access to the database.
- Concurrent schema transactions are not allowed.
- During the short period of time that the schema change transaction is in process, new database logins and schema readers will be blocked until the transaction completes.

Options to change the schema fields

You can change fields in the `_File` and `_Field` schema tables using any one of the following options:

- Make the change in a development database and then load the `delta.df` file into the schema online using the data dictionary tool. Be sure to check the **Add new objects on-line** box.
- Use `ALTER TABLE` or `ALTER COLUMN` (in OpenEdge SQL)
- Update the schema directly using `ABL`, as in the example below

Example

The following is an example of changing the label of the column to `Customer Number` using ABL:

```
SESSION:SCHEMA-CHANGE = “NEW OBJECTS”.
FIND FIRST _File WHERE _File-name='customer'.
FIND FIRST _Field WHERE _File-Recid = RECID(_File)
and _Field-Name='custNum'.
ASSIGN _Col-Label='Customer Number'.
```

The following is an example of changing the label of the `custNum` column to `Customer Number` using SQL:

```
ALTER TABLE Pub.Customer ALTER COLUMN CustNum SET PRO_COL_LABEL ‘Customer Number’;

ALTER TABLE Pub.Customer ALTER COLUMN CustNum SET PRO_COL_LABEL ‘Customer Number’;
```

Notes

- Non-structural fields do not affect the structure of the associated database tables when changed.

Change Int to Int64 online

You can change the datatype of a field in an application table from `Int` to `Int64` while the database remains running. Such a change does not require an exclusive schema lock.

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.

Effect on ABL applications

You only need to recompile `r-code` if the affected fields contain values that exceed the `Int` range.

For example, an application that uses `Buffer-Copy`, `Buffer-Compare`, or `Raw` transfer statements needs to be recompiled if the changed field contains values greater than the `Int` range after the datatype has been changed to `Int64`.

Users might consider changing `_Format` of the affected fields to reflect any changes.

The `Int` to `Int64` datatype change affects the cyclic redundancy check (CRC) of the table which contains the affected fields, but the `AVM` adapts to these differences and executes `r-code` normally. In other words, the AVM will execute the old`r-code` without requiring that it be recompiled.

Options to change the schema fields

You can change fields in the `_File` and `_Field` schema tables using any one of the following options:

- Make the change in a development database and then load the `delta.df` file into the schema online using the data dictionary tool. Be sure to check the **Add new objects on-line** box.
- ALTER TABLE or ALTER COLUMN (in OpenEdge SQL)
- ABL (by updating the schema directly, as in the example below.)

Examples

The following is an example of changing the datatype of the `custNum` column in the `Customer` table using ABL:

```
FIND FIRST _File WHERE _File-name='customer'.
FIND FIRST _Field WHERE _File-Recid = RECID(_File) and _Field-Name='custNum'.
ASSIGN _data-type='int64'.
ASSIGN _Format='->>,>>>,>>>,>>>,>>>'
```

The following is an example of changing the datatype of the `custNum` column in the `Customer` table using SQL:

```
ALTER TABLE Pub.Customer ALTER COLUMN CustNum SET PRO_DATA_TYPE BIGINT;
```

Change default storage areas for created objects while the database is running

By default, ABL, SQL, and the Data Dictionary create new database objects in the next storage area after the schema area.

To take advantage of the performance capabilities of Type II areas, you can specify a default area for the creation of tables, LOBs, and indexes, so that they are created in a Type II area instead of a Type I area like the schema area. You may also change the default object creation area at any time, whether or not the database is running.

Note: User-specified detault storage areas must be Type II areas.

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

For more information on configuring default areas, see "PROUTIL SETOBJECTDEFAULTAREA" in *Manage the OpenEdge Database*.

For details about showing default areas, see "PROUTIL SETOBJECTDEFAULTAREA" in *Manage the OpenEdge Database*.

To assign or change default storage areas for database objects, see these SQL statements in the *OpenEdge SQL Referrence*:

- ALTER TABLE
- CREATE INDEX
- CREATE TABLE

Note: Default storage area configuration and changes are currently unsupported at any time for objects created in schema holder databases for foreign databases (Data Servers), partitioned tables, or tenant-specific syntax for multi-tenant databases. You cannot set an encryption area as the default storage area.

Truncate and move tables while the database is running

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.

The PROUTIL utility lets you truncate and optionally move a table while the database is running, without having to recreate the schema.

See "PROUTIL TABLEMOVE qualifier" in *Manage the OpenEdge Database*.

4

Optimize database client notifications

To prevent unnecessary network traffic and improve response time, notifications of certain schema changes are automatically sent to remote clients.

You can use startup parameters, the PROMON utility, and VSTs to control the method and frequency of notification delivery.

To check the current status of notifications, check the columns **Complete Seq** and **In Progress Seq** on the User Notification Processes screen in the PROMON Processes/Clients Status display.

For more information, see 'Processes/Clients' in *OpenEdge Data Management: Database Administration*.

The _Connect VST fields associated with these columns are `_Connect-NotifySeq` and `_Connect-NotifySeqChecked`.

For more information, see 'Database connection (_Connect)' in *OpenEdge Data Management: Database Administration*.

Startup parameters that control notifications are:

- Database Notification Operations (`-dbnotifyops`) – sets the maximum number of concurrent notifications.
- Database Notification Time (`-dbnotifytime`) – controls how often a 12.2 client polls for notifications. If set, notifications are sent directly to clients, and clients incur no network traffic checking for them.
- Database Client Notification (`-usernotifytime`) – if set to non-zero value, specifies how often a self-service client polls the database to see if schema changes are in progress, or how often a remote client asks the server for notifications. For less frequent polling, specify a large number. This parameter generally affects only clients prior to version 12.2, but also affects 12.2 clients if you set `-dbnotifytime` to 0.

You can change the settings of these parameters while the database is online.

5

Manage brokers for continuous operation

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 from PROMON**—A PROMON option lets you disconnect secondary brokers without issuing a process kill signal.

  This option reduces potential database downtime. For details, 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 startup parameters for secondary brokers.

  Although you can list broker parameters for running servers by using PROMON R&D Status or Virtual System Tables, the database log file lets you troubleshoot landscape 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*.

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.

6

Enable automatic database reconnection

Use database parameters to define a *connection set* consisting of the primary database and up to two alternate databases, plus an optional retry count and pause interval.

You can specify your connection set using session startup parameters, in the `CONNECT` statement, or in a parameter file (`.pf`).

Specify a connection set using database connection parameters

The following database connection parameters specify connection set properties.

Table 3: Automatic connection properties

Parameter	Description
Alternate Database 1 (-dbalt1)	Connection information for the alternate database.
Alternate Database 2 (-dbalt2)	Connection information for the second alternate database, if there are multiple alternates.
Auto Reconnect (-autoReconnect)	For GUI or character-mode clients, restablishes connections to any disconnected databases specified through startup parameters.
Retry Connect Set (-retryConnect)	Number of retry attempts allowed for the connection set as a whole.
Retry Connect Set Pause (-retryConnectPause)	Interval to wait, in seconds, before retrying the connection set.

`-autoReconnect` is only for GUI or character-mode clients and indicates that if an unhandled STOP condition occurs, you want to try reconnecting databases connected using startup parameters before rerunning the `-p` procedure.

Database connection parameter considerations

- Like other database startup parameters, `-dbalt1`, `-dbalt2`, `-retryConnect`, and `-retryConnectPause` apply only to the last database parameter (-db) encountered in the parameter list thus far.
- You must specify the primary database first, before listing an alternate, otherwise the AVM issues an error.
- If both `-dbalt1` and `-dbalt2` are specified and the connection to the primary database fails, `-dbalt1` is tried first and `-dbalt2` is tried second.
- If you only specify `-dbalt2`, it becomes the first alternate.
- If the AVM is unable to connect to any database the `CONNECT` statement goes through the usual error handling.
- Any database parameters from the primary database, except for `-H` and `-S`, are also applied to the alternate databases, such as `-ld`, `-U` and `-P`.
- The logical name of the alternate database is always taken from the primary database, even if not explicitly set with `-ld`.
- If the same parameter is encountered more than once within the argument list for `-dbalt1` or `-dbalt2`, the last one is used. This is consistent with the parsing rules for parameters in general.
- If multiple `-dbalt1` or `-dbalt2` parameters are encountered within the parameter list for a particular primary database, the last one is used.
- `-ct` (Connection Retry Attempts) is an independent, but related, startup parameter that controls the number of connection retries for an *individual* database connection. The connection attempts are done in succession on the same database connection and have a default value of 50 attempts; the AVM pauses one second between connection attempts. This differs from `-retryConnect`, which applies to the connection set as a whole. If using both `-ct` and `-retryConnect`, you may want to set `-ct` to 0, or to a value lower than the default.

Example configurations

Example: Specify a primary database with one alternate:

```
-db mainDb -H PrimarySrvr -S 6999
       -dbalt1 'altDb -H AltSrvr1 -S 7999'
       -retryConnect 3
```

In this example the AVM would:

1. Try to connect to `mainDb`.
2. If that fails, it would try to connect to `altDb`.
3. If that fails, it would pause 5 seconds (default for `-retryConnectPause`) and repeat from step 1, three more times (because `-retryConnect` is 3).

Example: Specify a primary database with two alternates:

```
-db myDb -H PrimarySrvr -S 6999
       -dbalt1 'myDbTarget1 -H AltSrvr1 -S 7999'
       -dbalt2 'myDbTarget2 -H AltSrvr2 -S 7999'
       -retryConnect 10 -retryConnectPause 4
```

Example: Specify multiple primary databases with alternates:

```
-db myDb -H PrimarySrvr -S 6999
       -dbalt1 'myDbTarget1 -H AltSrvr1 -S 7999'
       -dbalt2 'myDbTarget2 -H AltSrvr2 -S 7999'
       -retryConnect 10 -retryConnectPause 4
       -db otherDb -H OtherDbSrvr -S 6998
       -dbalt1 'otherDbTarget1 -H AltOtherSrvr1 -S 7998'
       -dbalt2 'otherDbTarget2 -H AltOtherSrvr2 -S 7998'
       -retryConnect 8 -retryConnectPause 3
```

Usage of parameters in the CONNECT statement

You may also use these database connection parameters in a `CONNECT` statement. For `-dbalt1` and `-dbalt2` each alternate database name, along with its host and port parameters, must be enclosed in quotes. In the following example, the arguments to `-dbalt1` and `-dbalt2` are surrounded by single quotes and the connection string is surrounded by double quotes. You can reverse this and use double quotes to surround the inner arguments and use single quotes for the outer connection string.

In the following example code, one alternate database is specified:

```
CONNECT -db myDb -H PrimarySrvr -S 6999 -dbalt1 'myDbBackup -H AltSrvr1 -S 7999'.
```

The following example code assigns the connection string to a CHARACTER variable. It uses single quotes to enclose the inner `-dbalt1` parameters and double quotes for the outer string:

```
conn = "-db myDb -H PrimarySrvr -S 6999 -dbalt1 'myDbBackup -H AltSrvr1 -S 7999'".
       CONNECT VALUE(conn).
```

The following example code specifies two alternate databases:

```
conn = "-db myDb -H PrimarySrvr -S 6999
       -dbalt1 'myDbBackup -H AltSrvr1 -S 7999'
       -dbalt2 'myDbBackup -H AltSrvr2 -S 7999'
       -retryConnect 10 -retryConnectPause 4".
       CONNECT VALUE(conn).
```

Enable logging for troubleshooting

The `DB.Connects` log entry type (`-logentrytypes DB.Connects`) turns on logging for database connections. (For information on configuring and enabling logging, see 'Logging infrastructure overview' in *Troubleshoot ABL Applications*.) Normally, connect and disconnect messages are logged at the basic level (2 or higher). With this feature, additional messages are logged if the initial attempt to connect to the primary database fails. Consider the following connection string:

```
-db prim -H localhost -S 1234
       -dbalt1 "alt1 -H localhost -S 2345"
       -dbalt2 "alt2 -H localhost -S 3456"
       -retryConnect 2
       -retryConnectPause 3
```

At logging level 2 or higher, when there is a successful connection to an alternate database, a message is logged indicating the connection is to an alternate:

```
CONN           Connected to alternate database alt2
```

At logging level 3 or higher, additional messages are logged showing the connection attempts:

```
CONN           Attempting to connect to database prim (primary)
        CONN           Failed to connect to database prim
        CONN           Failover: attempting to connect to database alt1 (-dbalt1)
        CONN           Failed to connect to database alt1
        CONN           Failover: attempting to connect to database alt2 (-dbalt2)
```

At logging level 3 or higher, an additional message showing the pause between connection attempts, is logged before the connection set as a whole is tried again. The pause length is controlled by the value set for `-retryConnectPause`.

```
CONN           Pausing 3 seconds between connection attempts
```

At logging level 3 or higher, if the session is restarted due to an unhandled STOP condition, for a GUI or character-mode client, and `-autoReconnect` is enabled, a message is logged:

```
CONN           Session restart: attempting to reconnect any disconnected databases
```

Note: Single-user or self-service (shared-memory) connections do not support this feature.

See also

'Connection Retry Attempts (-ct)' in *Startup Command and Parameter Reference*

'CONNECT statement' in *ABL Reference*

'Database connection logging' in *Troubleshoot ABL Applications*

If a database is offline or otherwise unavailable, client-server connections can automatically connect to an alternate database, such as a replication target or backup database.

7

Manage OpenEdge replication for continuous operation

Validate database replication properties and run status

The DSRUTIL properties validation utility enables you to validate the replication properties of replication-enabled source and target databases and confirm running status of target database brokers.

VSTs help you manage data replication

Virtual system tables provide information to help you manage data replication. Examine fields in virtual system tables to determine the replication status of your databases.

Fields in virtual system tables offer a view into your replication landscape. For example, you'll find the current role for a database (source or target) in the Database status VST and server state details in the Replication VST.

Brokers run throughout recovery transitions

To finish transitions faster, database brokers continue running on single targets and replication sets. This improves performance for both manual and automatic recovery transitions. For more information, see 'Plan for transition' in *Use Database Replication*.

For details, see the following topics:

- OpenEdge Replication DSRUTIL Properties Validation Utility
- _Repl-Server VST field descriptions
- Database status (_DbStatus)

OpenEdge Replication DSRUTIL Properties Validation Utility

The DSRUTIL properties validation utility enables you to validate the replication properties of replication-enabled source and target databases and confirm running status of target database brokers.

When run against the replication source database, the utility performs the same validation made during startup of the replication server, validating all properties listed in the `[server]`, `[control-agent]`, and `[transition]` sections of the properties file. For each control-agent listed in the `[server].control-agents` property, the utility also pings the target database on the hostname and port in the associated `[control-agent]` section of the properties file.

When run against the replication target database, the utility performs the same validation made during startup of the replication agent, validating all properties listed in the `[agent]` and `[transition]` sections of the properties file. If there is a `[server]` section defined in the target database's properties file, the length of each control-agent name in the `[server].control-agents` property is validated.

The utility writes all properties file errors to both standard out and the database log file.

The utility can be run while the database is online or in single-user mode.

Syntax

The command for the utility is as shown:

```
dsrutil db-name -C validate
```

Parameters

db-name

The name of the database to perform the action on. The name of the database must be the first argument and must be a valid name.

Notes

- The utility adds the ability for the source database to report on the running status of any target database brokers using messages 19460 and 19461.

```
Database db-name for agent agent-name is online. Broker is running on
hostname:port-num. 19460
```

```
Unable to contact database db-name on hostname:port-num for agent agent-name.
19461
```

_Repl-Server VST field descriptions

The table below contains the field descriptions of the OpenEdge Replication virtual system table, `_Repl-Server`.

Table 4: Virtual system table _Repl-Server field description

Field name	Field number	Data type	Description
`_ReplSrv-id`	2	INT64	Server Number Format ->>>>>>>>>>>>>>>>>>9
`_ReplSrv-AgentCount`	3	INTEGER	Agents Count Number of agents that the OpenEdge Replication server is controlling Format >>>>>>>>>9
`_ReplSrv-BlocksSent`	4	INTEGER	After-image(AI) Blocks Sent Number of AI Blocks the OpenEdge Replication server has sent to connected OpenEdge Replication agents Format >>>>>>>>>9
`_ReplSrv-StartTime`	5	CHARACTER	Server Start Time Date and time when the OpenEdge Replication server was started Format X(24)
`_ReplSrv-LastBlockSentAt`	6	CHARACTER	Last Block Sent At Date and time when the OpenEdge Replication server was started Format X(24)
`ReplSrv-Database`	7	CHARACTER	Database file name Format X(24)
`ReplSrv-DatabaseRole`	8	CHARACTER	Database role. Valid values are: • Replication source • Replication target Format X(8)

Field name	Field number	Data type	Description
ReplSrv-ServerState	9	CHARACTER	Server state. Valid values are: • Performing Initialization • Connecting to Agents • Configuring Connected Agents • Performing Failure Recovery • Performing Startup Synchronization • In Normal Processing • Unknown Format X(36)
ReplSrv-DeferAgentConnUntil	10	CHARACTER	Date and time that deferred agent connection attempts will cease Format X(24)
ReplSrv-DeferAgentExpire	11	CHARACTER	Time remaining until deferred agent startup expires Format X(24)
ReplSrv-NextConnectIn	12	CHARACTER	Time until next connection attempt Format X(24)
ReplSrv-ConnectAttempts	13	INT64	Number of connection attempts performed Format ->>>>>>>>>>>>>>>>>>9
ReplSrv-AgentsConnected	14	INTEGER	Number of agents currently connected Format >>>>>>>>>9
ReplSrv-CurDelayInterval	15	INTEGER	Current message delay interval (in seconds) Format >>>>>>>>>9
ReplSrv-MinDelayInterval	16	INTEGER	Minimum message delay interval (in seconds) Format >>>>>>>>>9
ReplSrv-MaxDelayInterval	17	INTEGER	Maximum message delay interval (in seconds) Format >>>>>>>>>9

Field name	Field number	Data type	Description
ReplSrv-RecoveryState	18	CHARACTER	Server recovery state. Valid values are: • No Recovery • Just Entered Recovery • Server Attempting Connection • Initialize Synchronization with Agents • Synchronizing Agents • Recovery Complete Format X(42)
ReplSrv-RecAgentsToRecover	19	INTEGER	Number of agents needing recovery Format >>>>>>>>>9
ReplSrv-RecAgentsConnected	20	INTEGER	Number of agents connected and not needing recovery Format >>>>>>>>>9
ReplSrv-AgentsInSync	21	INTEGER	Number of agents in synchronization Format >>>>>>>>>9
ReplSrv-TransitionType	22	CHARACTER	Transition type. Valid values are: • Manual • Automatic Format X(32)
ReplSrv-TransitionTimeout	23	INTEGER	For automatic transition, the maximum amount of time (in seconds) that the target database will wait until it transitions to a normal database. Format >>>>>>>>>9
_ReplSrv-ReplicationSet	24	INTEGER	Replication set enabled Format >>>9

Field name	Field number	Data type	Description
_ReplSrv-TransDatabaseRole	25	CHARACTER	Role the database takes after transition. Valid values are: • Normal • Reverse Format X(32)
_ReplSrv-TransitionToAgents	26	CHARACTER	Transition to agents list Format X(100)
_ReplSrv-RestartAfterTrans	27	BOOLEAN	Restart after transition Format yes/no
_ReplSrv-AutoBeginAI	28	BOOLEAN	Automatically begin after-imaging during transition Format yes/no
_ReplSrv-AutoAddAI	29	BOOLEAN	Automatically add after-imaging files during transition Format yes/no
_ReplSrv-AIFile	30	CHARACTER	AI structure filename for automatically adding after-imaging Format X(256)
_ReplSrv-BackupMethod	31	CHARACTER	Backup method during transition Format X(32)
_ReplSrv-ConfiguredAgents	32	CHARACTER	List of configured agents Format X(68)
_ReplSrv-ReplKeepAlive	33	INT64	Replication keep alive property (in seconds) Format ->>>>9
_ReplSrv-SchemaLockAction	34	CHARACTER	Method server uses to perform schema locking. Valid values include: • Force • Wait Format X(32)

Field name	Field number	Data type	Description
_ReplSrv-AgentShutdownAction	35	CHARACTER	Action replication agent takes if connection to server is lost. Valid values include: • Normal • Recovery • Wait Format X(32)
_ReplSrv-ReservedChar	36	CHARACTER	Reserved
_ReplSrv-ReservedInt	37	INTEGER	Reserved
_ReplSrv-TransEndTrigger	38	INTEGER	Indicates if a transition end trigger is to execute on transition. Valid values include: • **0** — Execute a transition end trigger • **1** — Do not execute a transition end trigger Format >>>9
_ReplSrv-DRTransitionToAgents	39	CHARACTER	Recovery transition to agents list. Format X(100)
_ReplSrv-TransToDBName	40	CHARACTER	Remote Database Name Fully qualified database name to which the OpenEdge Replication agent is connected. Target database for the primary transition to agent in the event of failover transition. Format X(128)

Field name	Field number	Data type	Description
_ReplSrv-TransToDBHost	41	INTEGER	Remote Host Name The hostname of the primary transition to target database identified by the _replserv-TransToDBName field. Format X(128) **Note:** The hostname is the IP address instead of the hostname specified in repl properties or the startup (-H) on the target.
_ReplSrv-TransToDBPort	42	INTEGER	Connected to Agent on Port Number The port number the primary transition to target database identified by the _replserv-TransToDBName field as specified by the -S startup parameter. Format >>>>>>>>>9
_ReplSrv-SSLEnabled	43	BOOLEAN	Specifies whether or not SSL use is enabled for the replication database. • 0—SSL use is disabled. This is the default. • 1—SSL use is enabled. The property affects all replication processes that may run against the replication database.

Field name	Field number	Data type	Description
_ReplSrv-SSLNoHostVerify	44	BOOLEAN	Specifies whether or not the client verifies the SSL server's machine name. During the SSL handshake process, the SSL client may want to verify that the machine name where the SSL server runs matches what is in the server's certificate. • 0—Enables verification. • 1—Disables verification. In OpenEdge Replication, it is not guaranteed that the SSL server's host machine name will match what is stored in the server's certificate. For this reason, it is a best practice to disable this property.
_ReplSrv-SSLCertStorePath	45	CHARACTER	For an SSL client, the absolute path of the directory containing the SSL certificate for OpenEdge Replication SSL.
_ReplSrv-SSLCipherSuites	46	CHARACTER	A comma-separated list of the SSL cipher suites that can be used for any OpenEdge Replication SSL network connections for the replication database. Cipher suites are: • AES128-SHA256 • AES256-SHA256 • DHE-RSA-AES256-GCM-SHA384
_ReplSrv-SSLProtocols	47	CHARACTER	A comma-separated list of the SSL protocol versions that can be used for any OpenEdge Replication SSL network connections for the replication database. Supported SSL protocols are: • TLSv1.1 • TLSv1.2 • TLSv1.3

Note

- If the replication server is not running, the _replsrv VST returns no records.

Database status (_DbStatus)

The _DbStatus VST displays a wide variety of status data.

Field name	Data type	Description
_DbStatus-AiBlkSize	INTEGER	Number of bytes in an after-image block.
_DbStatus-BiBlkSize	INTEGER	Number of bytes in a before-image block.
_DbStatus-BiClSize	INTEGER	Number of kilobytes in the before-image cluster.
_DbStatus-BiOpen	CHARACTER	Most recent BI file open.
_DbStatus-BiSize	INT64	Logical size of the BI file. You can use this in conjunction with the _DbStatus-BiClSize value to determine the number of clusters in the BI file.
_DbStatus-BiTrunc	CHARACTER	Time since the last truncate.
_DbStatus-CacheStamp	CHARACTER	Schema cache file time stamp.
_DbStatus-Changed	INTEGER	Flag indicating if database has changed since last backup.
_DbStatus-ClVersMinor	INTEGER	Client minor version number.
_DbStatus-CreateDate	CHARACTER	Date and time of database creation.
_DbStatus-databaseRole	CHARACTER	Database role. Valid values are: • Replication source • Replication target • Normal **Note:** If replication is not enabled, the database role will be reported as Normal. Format X(8)
_DbStatus-DbBlkSize	INTEGER	Database block size.
_DbStatus-DbVers	INTEGER	Database version number.
_DbStatus-DbVersMinor	INTEGER	Database minor version number.

Field name	Data type	Description
_DbStatus-EmptyBlks	INT64	Number of empty blocks in the database.
_DbStatus-FbDate	CHARACTER	Most recent full backup.
_DbStatus-FreeBlks	INT64	Number of free blocks in the database.
_DbStatus-HiWater	INTEGER	Database blocks high-water mark.
_DbStatus-IbDate	CHARACTER	Most recent incremental backup.
_DbStatus-IbSeq	INTEGER	Sequence of last incremental backup.
_DbStatus-InitiateDiag	LOGICAL	Set to begin diagnostic data collection immediately
_DbStatus-Integrity	CHARACTER	DbStatus-Integrity enabled flag (`-i`).
_DbStatus-IntFlags	INTEGER	Integrity flags.
_DbStatus-LastLgTruncateTime	CHARACTER	Date-time of the most recent truncation of the database log file
_DbStatus-LastOpen	CHARACTER	Most recent database open.
_DbStatus-LastTable	INTEGER	Highest table number defined.
_DbStatus-LastTran	INTEGER	Last transaction ID.
_DbStatus-MostLocks	INTEGER	Lock table high-water mark.
_DbStatus-NumAreas	INTEGER	Number of areas.
_DbStatus-NumLocks	INTEGER	Lock table entries in use.
_DbStatus-NumSems	INTEGER	Number of semaphores.
_DbStatus-PrevOpen	CHARACTER	Previous database open.
_DbStatus-RestrictMode	CHARACTER	The state of any access restrictions set by PROUTIL DBRESTRICT.
_DbStatus-RMFreeBlks	INT64	Number of rm blocks with free space.
_DbStatus-SharedMemVer	INTEGER	Shared memory version number.
_DbStatus-ShmVers	INTEGER	Shared memory version number.
_DbStatus-Starttime	CHARACTER	Time when the database was started.
_DbStatus-State	INTEGER	State of the database.

Field name	Data type	Description
_DbStatus-Tainted	INTEGER	Damaged database flags.
_DbStatus-TotalBlks	INT64	Number of blocks allocated to the database.
_DbStatus-ZeroStatsDate	CHARACTER	Date/time of the last PROUTIL ZEROSTATS operation on the database.

8

Update ABL client applications

To support continuous operation, built-in OpenEdge database features allow ABL client application updates while the database is running. You can change trigger definitions or rename user table fields without requiring exclusive schema locks, and renaming fields does not disconnect users from the database.

For details, see the following topics:

- Change ABL triggers online
- Rename fields using ABL
- Drop a field online

Change ABL triggers online

ABL trigger definitions can be changed online. That means you can create, change, or delete ABL triggers without an exclusive schema lock.

How ABL clients handle an ABL trigger change

When an ABL trigger is changed online, all the connected ABL clients are notified of the change. This functionality is specific to ABL clients in OpenEdge 12.2.

When a trigger change is initiated, the ABL client that executes the change checks whether earlier versions of the ABL client are connected to the database. If an earlier version is found, then either the transaction terminates or is blocked until all earlier ABL clients are disconnected from the database.

Check the ABL client version

Progress recommends that you check the ABL client versions before changing ABL triggers online.

The ABL client version information is kept in the _Connect virtual system table (VST). Use one of the following methods to determine the version of an ABL client connection:

- Run the following ABL query:

```
for each _Connect where _Connect-NotifyVers < 1
      and _Connect-Name <> ?::
      display _Connect-Name _Connect-NotifyVers.
```

- Use PROMON:

  1. Launch Proenv.
  2. At the command line, type promon *database* and press ENTER..

     The OpenEdge monitor is displayed.
  3. Type **R&D** and press ENTER.

     The R&D menu is displayed.
  4. Type **1** and press ENTER.

     The status monitoring options are displayed.
  5. Type **4** and press ENTER.

     Process and client information is displayed.
  6. Type **9** and press ENTER.

     The User Notification Processes table is displayed.
  7. Review the values in the Notify Version column.

     A 1 means that the client supports online changes to triggers. A 0 means that the client does not support online changes to triggers.
  8. Close all the client connections that do not support online changes to triggers.
  9. Create, change, or delete an ABL trigger online.

Note: You may want to set the -dbnotifytime database startup parameter to a smaller value. For example, if the parameter is currently set to the default value of 30 seconds, decrease the time. Setting the parameter to a smaller value enables clients to learn about trigger changes sooner.

How changing an ABL trigger online affects ABL clients

After the ABL client that executed a trigger change commits the change, the client notifies all the other connected ABL clients that an ABL trigger change occurred. Then, the connected ABL clients complete the database transactions that are in process. When the transactions complete, the clients refresh their cache. When the next trigger event occurs, the ABL virtual machine (AVM) applies the new trigger definition. If a trigger is deleted, the events associated with the trigger no longer occur.

Because a trigger change can affect multiple database tables and fields, an ABL client is not aware of the change until the transaction is committed.

Effect on ABL applications

You do not need to recompile ABL code that uses changed trigger definitions in order for online changes to take effect.

Effects on transactional concurrency

Changing ABL triggers online has the following effects on the concurrency of database transactions:

- Concurrent schema changes are not allowed.
- New database connection requests are not allowed while a trigger change is in process. These requests must be terminated or wait until the transaction completes.
- Concurrent transactions cannot read schema data. Read operations fail or are blocked until the trigger change completes.

Methods for changing ABL triggers online

Use one of the following options to change an ABL trigger online:

- Data Administration—Select the **Add new objects on-line** checkbox in the Load Data Definitions dialog box to perform the loading operation online. Use this option only to load a delta `.df` file when there is a difference between triggers on two databases (a source database and a target database).

  For more information, see 'Navigate within the Data Administration tool in Windows' in *Database Administration Tools*.

- ABL—Directly update the corresponding schema records and columns (which contain the trigger definitions). The AVM manages these changes internally.

  The following ABL procedure creates a new ABL trigger. The new trigger runs the `create.p` program whenever a record is created in the `Item` table:

```
/* Fetches Table record id from _File System table */
      FIND _File WHERE _File-name = 'Item'.
      /* Creating record in _File-Trig */
      CREATE _File-trig.
      _File-trig._File-recid = RECID(_File).
      _File-trig._event = 'CREATE'.
      _File-trig._proc-name = 'create.p'.
      _File-trig._override = YES.
```

Rename fields using ABL

To support continuous operation, you can rename user table fields while the database is running. Learn how to rename fields and how field renaming affects transactional concurrency and connected clients.

Renaming a database field no longer requires an exclusive schema lock or disconnecting users from the database. When a user table field is renamed, all connected ABL clients are notified of the change.

This functionality is specific to ABL clients in this release of OpenEdge. When a field is renamed, the ABL client that executes the change checks to see if any earlier versions of the ABL client are connected to the database. If the client finds an earlier version, then the process is treated as an offline operation.

How renaming a field affects ABL clients

After the ABL client renames a field on an active database, a notification is sent to all the other connected clients that a field has been renamed. The connected clients then complete the database transactions in process. When the transactions are completed, the clients can refresh their cache.

Effects on transactional concurrency

Renaming a field online has the following effects on the concurrency of database transactions:

- Concurrent schema changes are not allowed.
- New database connection requests are not allowed while a field rename is in process. These requests must be terminated or wait until the transaction completes.
- Concurrent transactions cannot read schema data. Read operations fail or are blocked until the rename change completes.

Methods for renaming fields online

Use one of the following options to rename a field:

- Data Administration—Select the **Add new objects online** checkbox in the Load Data Definitions dialog box to perform the loading operation online. Use this option only to load a delta `.df` file when there is a difference between two databases (a source database and a target database).

  For more information, see 'Navigate within the Data Administration tool in Windows' in *Database Administration Tools*.

- ABL—Directly update the corresponding schema records. The AVM manages these changes internally.

  Use the following ABL procedure to rename a field.

```
FIND FIRST _File WHERE _File-name = 'table name'.
      FIND FIRST _Field WHERE _Field._Field-Name='column name'
      AND _Field._File-recid = RECID(_File).
      UPDATE _Field._Field-Name='New name'.
```

Drop a field online

You can remove a named field from the schema definition while the database remains running and users are connected to the database. The data of the field in the database table is also effectively removed in the process without requiring any exclusive lock.

How dropping a field online affects ABL clients

During an online `Drop Field` transaction, concurrent ABL clients are prevented from accessing the field that is being dropped.

After an ABL client drops a field online, the client notifies all the other connected clients that a field is dropped. The connected clients then complete the database transactions that are in process. After a client completes a transaction, it refreshes its schema cache.

Effects on transactional concurrency

Dropping a field online has the following effects on the concurrency of database transactions:

- Concurrent schema changes are not allowed while a field drop transaction is in process.
- New database connection requests are not allowed while the field drop transaction is in process. These requests must be terminated or wait until the transaction completes.
- Concurrent transactions cannot read schema data. Read operations fail or are blocked until the field drop transaction is complete.

Methods for dropping a field online

You can drop a field online using the Data Adminstration tool. Select the **Add new objects online** checkbox in the Load Data Definitions dialog box to perform the loading operation online. Use this option only to load a delta `.df` file when there is a difference between two databases (a source database and a target database).

For more information, see 'Navigate within the Data Administration tool in Windows' in *Database Administration Tools*.

Limitations of dropping fields online

Some limitations of dropping fields online are as follows:

- 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.