PROGRESS® OpenEdge®

OpenEdge® Deployment: Startup Command and Parameter Reference

PROGRESS

Notices

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.

Business Making Progress, Corticon, DataDirect (and design), DataDirect Cloud, DataDirect Connect, DataDirect Connect64, DataDirect XML Converters, DataDirect XQuery, Deliver More Than Expected, Easyl, Fathom, Icenium, Kendo UI, Making Software Work Together, OpenEdge, Powered by Progress, Progress, Progress Control Tower, Progress RPM, Progress Software Business Making Progress, Progress Software Developers Network, Rollbase, RulesCloud, RulesWorld, SequeLink, Sitefinity (and Design), SpeedScript, Stylus Studio, TeamPulse, Telerik, (Telerik (and Design), Test Studio, and WebSpeed are registered trademarks of Progress Software Corporation or one of its affiliates or subsidiaries in the U.S. and/or other countries. AccelEvent, AppsAlive, AppServer, BravePoint, BusinessEdge, DataDirect Spy, DataDirect SupportLink, , Future Proof, High Performance Integration, Modulus, NativeScript, OpenAccess, Pacific, ProDataSet, Progress Arcade, Progress Pacific, Progress Profiles, Progress Results, Progress RFID, Progress Progress Software, ProVision, PSE Pro, SectorAlliance, Sitefinity, SmartBrowser, SmartComponent, SmartDataBrowser, SmartDataObjects, SmartDataView, SmartDialog, SmartFolder, SmartFrame, SmartObjects, SmartPanel, SmartQuery, SmartViewer, SmartWindow, WebClient, and Who Makes Progress 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.

Please refer to the Release Notes applicable to the particular Progress product release for any third-party acknowledgements required to be provided in the documentation associated with the Progress product.

The Release Notes can be found in the OpenEdge installation directory and online at:
https://community.progress.com/technicalusers/w/openedgegeneral/1329.openedge-product-documentation-overview.aspx.

For the latest documentation updates see OpenEdge Product Documentation on Progress Communities: (https://community.progress.com/technicalusers/w/openedgegeneral/1329.openedge-product-documentation-overview.aspx).

October 2015

Last updated with new content: Release 11.6.0

Table of Contents

Preface

For details, see the following topics:

Purpose

This book provides a comprehensive reference to the startup and shutdown commands and startup parameters for OpenEdge®.

Audience

The primary audiences for this book are OpenEdge business application developers and system administrators. The secondary audiences are OpenEdge developers and technical support engineers.

Organization

Introduces the commands and parameters you can use to start up and shut down OpenEdge client processes. It also describes how to use a parameter file (`.pf`) to maintain startup parameters for a particular database startup, client session startup, or database connection.

Describes the OpenEdge client startup commands that you can enter at the command line. It describes the syntax of each client startup command, summarizes the tasks you can perform with the startup commands, and then describes each command and its parameters in detail.

Provides an overview of the OpenEdge startup parameters organized by usage category.

Provides a detailed reference description for each of the OpenEdge startup parameters, organized in alphabetical order by syntax.

Using this manual

OpenEdge provides a special purpose programming language for building business applications. In the documentation, the formal name for this language is *ABL (Advanced Business Language)*. With few exceptions, all keywords of the language appear in all `UPPERCASE`, using a font that is appropriate to the context. All other alphabetic language content appears in mixed case.

For the latest documentation updates see the OpenEdge Product Documentation Overview page on Progress Communities:

https://community.progress.com/technicalusers/w/openedgegeneral/1329.openedge-product-documentation-overview.aspx .

References to ABL compiler and run-time features

ABL is both a compiled and an interpreted language that executes in a run-time engine. The documentation refers to this run-time engine as the *ABL Virtual Machine (AVM)*. When the documentation refers to ABL source code compilation, it specifies *ABL* or *the compiler* as the actor that manages compile-time features of the language. When the documentation refers to run-time behavior in an executing ABL program, it specifies *the AVM* as the actor that manages the specified run-time behavior in the program.

For example, these sentences refer to the ABL compiler's allowance for parameter passing and the AVM's possible response to that parameter passing at run time: "ABL allows you to pass a dynamic temp-table handle as a static temp-table parameter of a method. However, if at run time the passed dynamic temp-table schema does not match the schema of the static temp-table parameter, the AVM raises an error." The following sentence refers to run-time actions that the AVM can perform using a particular ABL feature: "The ABL socket object handle allows the AVM to connect with other ABL and non-ABL sessions using TCP/IP sockets."

References to ABL data types

ABL provides built-in data types, built-in class data types, and user-defined class data types. References to built-in data types follow these rules:

- Like most other keywords, references to specific built-in data types appear in all `UPPERCASE`, using a font that is appropriate to the context. No uppercase reference ever includes or implies any data type other than itself.
- Wherever *integer* appears, this is a reference to the `INTEGER` or `INT64` data type.
- Wherever *character* appears, this is a reference to the `CHARACTER`, `LONGCHAR`, or `CLOB` data type.
- Wherever *decimal* appears, this is a reference to the `DECIMAL` data type.
- Wherever *numeric* appears, this is a reference to the `INTEGER`, `INT64`, or `DECIMAL` data type.

References to built-in class data types appear in mixed case with initial caps, for example, `Progress.Lang.Object`. References to user-defined class data types appear in mixed case, as specified for a given application example.

Typographical conventions

This manual uses the following typographical and syntax conventions:

Convention	Description
Bold	Bold typeface indicates commands or characters the user types, provides emphasis, or the names of user interface elements.
Italic	Italic typeface indicates the title of a document, or signifies new terms.
SMALL, BOLD CAPITAL LETTERS	Small, bold capital letters indicate OpenEdge key functions and generic keyboard keys; for example, **GET** and **CTRL**.
KEY1+KEY2	A plus sign between key names indicates a **simultaneous** key sequence: you press and hold down the first key while pressing the second key. For example, **CTRL+X**.
KEY1 KEY2	A space between key names indicates a **sequential** key sequence: you press and release the first key, then press another key. For example, **ESCAPE H**.
Syntax:	
`Fixed width`	A fixed-width font is used in syntax, code examples, system output, and file names.
`Fixed-width italics`	Fixed-width italics indicate variables in syntax.

Convention	Description
Fixed-width bold	Fixed-width bold italic indicates variables in syntax with special emphasis.
UPPERCASE fixed width	ABL keywords in syntax and code examples are almost always shown in upper case. Although shown in uppercase, you can type ABL keywords in either uppercase or lowercase in a procedure or class.
Period (.) or colon (:)	All statements except DO, FOR, FUNCTION, PROCEDURE, and REPEAT end with a period. DO, FOR, FUNCTION, PROCEDURE, and REPEAT statements can end with either a period or a colon.
[ ]	Large brackets indicate the items within them are optional.
[]	Small brackets are part of ABL.
{ }	Large braces indicate the items within them are required. They are used to simplify complex syntax diagrams.
{}	Small braces are part of ABL. For example, a called external procedure must use braces when referencing arguments passed by a calling procedure.
|	A vertical bar indicates a choice.
. . .	Ellipses indicate repetition: you can choose one or more of the preceding items.

Examples of syntax descriptions

In this example, ACCUM is a keyword, and *aggregate* and *expression* are variables:

Syntax

```
ACCUM aggregate expression
```

FOR is one of the statements that can end with either a period or a colon, as in this example:

```
FOR EACH Customer NO-LOCK:
  DISPLAY Customer.Name.
END.
```

In this example, STREAM *stream*, UNLESS-HIDDEN, and NO-ERROR are optional:

Syntax

```
DISPLAY [ STREAM stream ] [ UNLESS-HIDDEN ] [ NO-ERROR ]
```

In this example, the outer (small) brackets are part of the language, and the inner (large) brackets denote an optional item:

Syntax

```
INITIAL [ constant [ , constant ] ]
```

A called external procedure must use braces when referencing compile-time arguments passed by a calling procedure, as shown in this example:

Syntax

```
{ &argument-name }
```

In this example, `EACH`, `FIRST`, and `LAST` are optional, but you can choose only one of them:

Syntax

```
PRESELECT [ EACH | FIRST | LAST ] record-phrase
```

In this example, you must include two expressions, and optionally you can include more. Multiple expressions are separated by commas:

Syntax

```
MAXIMUM ( expression , expression [ , expression ] ... )
```

In this example, you must specify `MESSAGE` and at least one *`expression`* or `SKIP [ ( n ) ]`, and any number of additional *`expression`* or `SKIP [ ( n ) ]` is allowed:

Syntax

```
MESSAGE { expression | SKIP [ ( n ) ] } ...
```

In this example, you must specify `{`*`include-file`*, then optionally any number of *`argument`* or `&`*`argument-name`*` = "`*`argument-value`*`"`, and then terminate with `}`:

Syntax

```
{ include-file
    [ argument | &argument-name = "argument-value" ] . . . }
```

Long syntax descriptions split across lines

Some syntax descriptions are too long to fit on one line. When syntax descriptions are split across multiple lines, groups of optional and groups of required items are kept together in the required order.

In this example, `WITH` is followed by six optional items:

Syntax

```
WITH [ ACCUM max-length ] [ expression DOWN ]
  [ CENTERED ] [ n COLUMNS ] [ SIDE-LABELS ]
  [ STREAM-IO ]
```

Complex syntax descriptions with both required and optional elements

Some syntax descriptions are too complex to distinguish required and optional elements by bracketing only the optional elements. For such syntax, the descriptions include both braces (for required elements) and brackets (for optional elements).

In this example, `ASSIGN` requires either one or more *`field`* entries or one *`record`*. Options available with *`field`* or *`record`* are grouped with braces and brackets:

Syntax

```
ASSIGN   { [ FRAME frame ] { field [ = expression ] }
             [ WHEN expression ] } . . .
       | { record [ EXCEPT field . . . ] }
```

OpenEdge messages

OpenEdge displays several types of messages to inform you of routine and unusual occurrences:

- **Execution messages** inform you of errors encountered while OpenEdge is running a procedure; for example, if OpenEdge cannot find a record with a specified index field value.

- **Compile messages** inform you of errors found while OpenEdge is reading and analyzing a procedure before running it; for example, if a procedure references a table name that is not defined in the database.
- **Startup messages** inform you of unusual conditions detected while OpenEdge is getting ready to execute; for example, if you entered an invalid startup parameter.

After displaying a message, OpenEdge proceeds in one of several ways:

- Continues execution, subject to the error-processing actions that you specify or that are assumed as part of the procedure. This is the most common action taken after execution messages.
- Returns to the Procedure Editor, so you can correct an error in a procedure. This is the usual action taken after compiler messages.
- Halts processing of a procedure and returns immediately to the Procedure Editor. This does not happen often.
- Terminates the current session.

OpenEdge messages end with a message number in parentheses. In this example, the message number is `200`:

```
** Unknown table name table. (200)
```

If you encounter an error that terminates OpenEdge, note the message number before restarting.

Obtaining more information about OpenEdge messages

In Windows platforms, use OpenEdge online help to obtain more information about OpenEdge messages. Many OpenEdge tools include the following Help menu options to provide information about messages:

- Choose **Help** > **Recent Messages** to display detailed descriptions of the most recent OpenEdge message and all other messages returned in the current session.
- Choose **Help** > **Messages** and then type the message number to display a description of a specific OpenEdge message.
- In the Procedure Editor, press the **HELP** key or **F1**.

On UNIX platforms, use the OpenEdge `pro` command to start a single-user mode character OpenEdge client session and view a brief description of a message by providing its number.

To use the pro command to obtain a message description by message number:

1. Start the Procedure Editor:

   ```
   OpenEdge-install-dir/bin/pro
   ```

2. Press **F3** to access the menu bar, then choose **Help** > **Messages**.
3. Type the message number and press **ENTER**. Details about that message number appear.
4. Press **F4** to close the message, press **F3** to access the Procedure Editor menu, and choose **File** > **Exit**.

1

Introduction

This chapter introduces the commands and parameters you can use to start up and shut down OpenEdge® client processes. The following sections describe the commands and parameters, as well as how to use a parameter (`.pf`) file to maintain startup parameters for a particular database startup, client session startup, or database connection.

For details, see the following topics:

- About startup commands
- About startup parameters
- Using parameter files

About startup commands

OpenEdge provides startup commands that support its client/server architecture and a wide range of configuration options. In client/server configurations, the client application and database server are separate processes. OpenEdge applications execute in a client session, sending requests to the OpenEdge server. The OpenEdge server accesses the database on behalf of each client session.

Consequently, the startup commands fall into two general categories:

- Commands to startup and shutdown database processes, such as database servers, database brokers, and background writer.
- Commands to start up and shutdown client sessions that connect to databases, such as an ABL (Advanced Business Language) client, AppServer™, and DataServer client.

After setting up your database, you are ready to run OpenEdge. In general, you start clients and servers as follows:

- Start a database server or broker for a database. Starting a server or broker for a database is distinct from connecting to a database. For more information on starting these processes, see *OpenEdge Getting Started: Installation and Configuration*.
- Start the client applications.

For more information about the startup commands, see Client Startup Commands on page 27.

For information about commands to startup and shutdown database processes, see *OpenEdge Data Management: Database Administration*.

About startup parameters

You use startup parameters with the startup commands to help define the context in which the database processes and client sessions run. The requirements and use of parameters varies with the different operating systems and network environments. For example, you can use the startup parameters to control how a client connects to a database and enhance database server and client performance. Startup parameters are organized within the following usage categories:

- Client session
- Client connection
- OpenEdge database server
- DataServer
- SQL

For more information about the startup parameter usage categories, see Startup Parameter Usage Categories on page 35. For details on the individual startup parameters, see Startup Parameter Descriptions on page 57.

You can access the list of startup parameters defined at startup for the current OpenEdge session by reading the `STARTUP-PARAMETERS` attribute on the `SESSION` system handle from an ABL procedure. For more information about the `STARTUP-PARAMETERS` attribute, or the `SESSION` system handle, see *OpenEdge Development: ABL Reference*.

Using parameter files

A *parameter file* is an operating system file that contains one or more startup parameters. The primary reason to use a parameter file is to avoid typing commands each time you want to start OpenEdge. Instead of repeatedly typing a series of parameters and other arguments, type them once and place them in a parameter file. You can also use a parameter file to maintain the startup parameters for a particular database, group of users, or system configuration. A parameter file has a `.pf` extension.

Every OpenEdge executable invokes a parameter file named `startup.pf` by default. This file is located in the `DLC` directory. It must exist for OpenEdge to execute properly. You can modify `startup.pf` to add parameter definitions. You can also create one or more additional parameter files and direct OpenEdge to invoke them after it invokes `startup.pf`.

Note: In Windows, you can specify the name of the OpenEdge startup file using the PROSTARTUP environment variable in the progress.ini file and the Registry. See *OpenEdge Deployment: Managing ABL Applications* for more information on the progress.ini file.

Calling parameter files

To call an additional parameter file, use the Parameter File (-pf) startup parameter in the OpenEdge startup command line or in a CONNECT statement.

Calling a parameter file in a startup command

From the startup command line, use the Parameter File (-pf) startup parameter to call the parameter file, as shown in the following table:

Operating system	Syntax
UNIX	`pro sports -pf` *`filename`*`.pf`
Windows	`prowin32 -1 sports -pf` *`filename`*`.pf` (Specify in the **Command Line** field of a **Shortcut Properties** dialog box.)

The *filename* can be any valid filename. OpenEdge combines all parameters on a command line with those in the parameter file. If the same parameter is mentioned more than once, the last occurrence takes precedence. For more information on startup parameters, see Startup Parameter Descriptions on page 57.

Calling a parameter file in a CONNECT statement

Use the following syntax to specify a parameter file in a CONNECT statement in an ABL procedure:

```
CONNECT -pf sports.pf
```

For example, the following new_york.p procedure connects using a parameter file called new_york.pf.

```
UPDATE ny_user ny_password BLANK.
  CONNECT -pf new_york.pf
    -U VALUE(ny_user) -P VALUE(ny_passwd)
NO-ERROR.
  IF NOT CONNECTED("new_york") THEN
    MESSAGE "Connection to New York office failed".
```

Calling multiple parameter files

You can use the `-pf` parameter as many times as you like on a command line. This lets you specify application-specific parameters in one parameter file, database-specific parameters in a second parameter file, and user-specific parameters in yet another parameter file. It also allows you to connect to multiple databases from the same command line or `CONNECT` statement.

When connecting to more than one database, the startup parameters must appear after the corresponding Physical Database Name (`-db`) parameter and database name. If connecting to more than one database, consider putting all the database-related parameters in a parameter file, then add the `-db` parameter to the top of the file. To support earlier releases, OpenEdge assigns any startup parameters that precede all database names to the first database on the command line.

OpenEdge processes parameters from left to right. Each `-pf` parameter encountered is processed first before the next top-level parameter is evaluated. If a parameter file specified by `-pf` contains another `-pf` parameter, the nested parameter file is evaluated before resuming the top-level processing.

Note: If the same parameter is mentioned more than once, the last occurrence takes precedence.

Creating and editing parameter files

You can use any text editor to create a file containing startup parameters. Follow the format explained in Parameter file format on page 25. You can also use OpenEdge tools to create and edit parameter files.

To create and edit parameter files:

1. Access the Data Administration tool or Data Dictionary.
2. Choose **Utilities** > **Editor for Parameter Files**

   OpenEdge prompts you for the file you want to create or edit.
3. Specify the filename.

   OpenEdge alphabetically lists all available startup parameters.
4. Specify which of the parameters you want in the file by choosing the parameter and entering `yes`, `no`, or the appropriate value.
5. Enter any comments you want to include in the parameter file.
6. After creating the parameter file, OpenEdge returns you to the Data Administration or Data Dictionary main window.

Parameter file format

The following `new_york.pf` file is a sample parameter file:

```
-db  /usr/disk0/newyork          # new york central db

-B   200                         # database buffers
-H   nysystem -S nyserver        # host, service
-L   10000                       # lock table entries

#    All other parameters receive their default values.
```

The format of the parameter file is the same for all operating systems. Follow these rules when creating a parameter file:

- Place a parameter and its argument on the same line.
- Use a maximum of 40 tokens per line in a parameter file.
- Specify parameters with the UNIX syntax.
- Use a pound sign (`#`) to begin a comment. OpenEdge ignores the rest of the line.
- Enclose a value in single (`'`) or double (`"`) quotes. Any white space inside the quotes is treated as part of the token.
- Any parameter not named in a parameter file receives a default value. To remind yourself of the default values, consider specifying all parameters and their values in the parameter file and adding the comment "`# default`" to those values that do not change.
- Use a tilde (`~`) to suppress the special meaning of the single (`'`) or double (`"`) quotes or tilde (`~`). A tilde also prefixes the control characters that OpenEdge can recognize in a parameter file, as listed in the following table.

Character	Description	Octal code
~*nnn*	The character having octal code *nnn*	—
~b	BS character	~010
~t	HT (horizontal tab) character	~011
~n	LF (line feed) character	~012
~r	VT (carriage return) character	~013
~f	FF (form feed) character	~014
~E	ESC character.	~033

On UNIX, use the backslash (`\`) in place of a tilde.

2

Client Startup Commands

This chapter describes the OpenEdge client startup commands that you can enter at the command line. It describes the syntax for each client startup command, summarizes the tasks you can perform with the startup commands, and then describes each command and its parameters.

For details, see the following topics:

Client startup command syntax

The conventions used in command syntax descriptions are as follows:

```
command [ db-name ] [ parameter ] [ value ] ...
```

For example, you can use the following command to allow 100 users to access the sports database and then set values for the database connection, performance, and network parameters:

```
prowin32 -db sports -n 100 -B 30000 -L 1000 -S sprtsv -H sys27
```

The following table describes each command component.

Component	Description
command	On UNIX, the command runs a script that executes an OpenEdge executable with appropriate parameters. In Windows, some commands run a batch file that executes an OpenEdge executable with appropriate parameters. Other commands run an OpenEdge executable directly.
db-name	Name of the database to which you want to connect.
parameter	Operating criteria for the command.
value	Numeric value or file specification for the parameter.

Observe the following conventions when entering a command:

- On UNIX, commands are lowercase. In Windows, commands are not case sensitive.
- On UNIX and Windows, enter parameters exactly as shown in the syntax descriptions.
- On UNIX, values can be case sensitive; for example, UNIX filenames are case sensitive. In addition, in Windows using the NT File System (NTFS), values can be case sensitive. In Windows using the File Allocation Table (FAT or FAT32) file system, values are not case sensitive.

Tasks you can perform with startup and shutdown commands

Startup commands startup and shut down OpenEdge database or client processes. The following table summarizes the tasks you can perform on UNIX and Windows systems and the related startup and shutdown commands.

Task	Command
Start a character Procedure Editor and connect to a single-user database.	`pro` *`db-name`*

Start a character Procedure Editor and connect to a multi-user database.	`mpro` *`db-name`*
	`-S` *`service-name`*
	`-H` *`host-name`*
	`-N` *`network-type`*

Start an OpenEdge character client session in batch mode and connect to a single-user database.	`bpro` *`db-name`* `-p` *`procedure-name`*

Start an OpenEdge character client session in batch mode and connect to a multi-user database.	`mbpro` *`db-name`* `-p` *`procedure-name`*
	`-S` *`service-name`*
	`-H` *`host-name`*
	`-N` *`network-type`*

The following table summarizes tasks and their related startup commands for Windows only. You can use these commands in the **Command Line** field of a **Shortcuts Properties** dialog box.

Task	Command
Start a GUI Procedure Editor and connect to a single-user database.	`prowin32 -1` *`db-name`*

Start a GUI Procedure Editor and connect to a multi-user database.	`prowin32` *`db-name`*
	`-S` *`service-name`*
	`-H` *`host-name`*
	`-N` *`network-type`*

Start the ADE Desktop and connect to a single-user database.	`prowin32 -p _desk.p -1` *`db-name`*

Start the ADE Desktop and connect to a multi-user database.	`prowin32 -p _desk.p -1` *`db-name`*
	`-S` *`service-name`*
	`-H` *`host-name`*
	`-N` *`network-type`*

Start an OpenEdge batch session and connect to a single-user database.	`prowin32 -b -1` *`db-name`* `-p` *`procedure`*

Start an OpenEdge batch session and connect to a multi-user database.	`prowin32 -b` *`db-name`*
	`-p` *`procedure`*
	`-S` *`service-name`*
	`-H` *`host-name`*
	`-N` *`network-type`*

Client startup command descriptions

This section contains a description of each of the following OpenEdge startup commands:

- BPRO command on page 31
- MBPRO command on page 31
- MPRO command on page 32
- PRO command on page 33
- PROWIN32 command (Windows only) on page 34

Note: In general, if the database is not in your current directory, you must specify the full pathname of the database. There are a few exceptions. See the description of the command you want to use for more information.

For information about commands to startup and shutdown database processes, see *OpenEdge Data Management: Database Administration*.

BPRO command

Starts a single-user OpenEdge client session in batch or background mode.

Syntax

```
bpro [db-name] -p procedure-name [parameters ][ > output-file]
```

Parameters

db-name

Specifies the database you want to connect.

-p *procedure-name*

Specifies a procedure to run at startup.

parameters

Specifies the OpenEdge startup parameters you want to use.

output-file

Specifies the name of the file where messages are sent.

Notes

- For UNIX, the `BPRO` command runs the following executable:

  ```
  _progres -1 -b &
  ```

  The `&` character causes an OpenEdge batch session to run in the background, and returns control to the terminal once the session starts. When the `&` character is not present (for example, when you simply use the (`-b`) startup parameter), OpenEdge initiates a batch session in the foreground without terminal interaction and control does not return to the terminal until the session completes.

- On UNIX and in Windows, you can redirect batch job input and output with the < and > redirection symbols. You also can use the pipe symbol (`|`) to put an OpenEdge batch run in a command pipeline.

MBPRO command

Starts a multi-user OpenEdge client session in batch or background mode.

Syntax

```
mbpro [db-name] -p procedure-name [parameters] [ > output-file]
```

Parameters

db-name

Specifies the database to which you want to connect. If you are connecting to a database through shared memory and the database is not in your current directory, you must specify the path of the database. On the other hand, if you are making a client/server connection to the database, you do not need to specify a path.

-p *procedure-name*

Specifies the procedure to run at startup.

parameters

Specifies the startup parameters you want to use.

output-file

Specifies a file where output messages are sent.

Notes

- On UNIX, the `MBPRO` command runs the following executable:

  ```
  _progres -b &
  ```

- On UNIX and in Windows, you can redirect batch job input and output with the < and > redirection symbols. You also can use the pipe symbol (|) to put an OpenEdge batch run in a command pipeline.

MPRO command

Starts a multi-user mode character OpenEdge client session.

Syntax

```
mpro [ db-name] [parameters]
```

Parameters

db-name

Specifies the database to which you want to connect. If you are connecting to a database through shared memory and the database is not in your current directory, you must specify the path of the database. On the other hand, if you are making a client/server connection to the database, you do not need to specify a path.

parameters

Specifies the startup parameters you want to use.

Notes

- The `MPRO` command runs the following executable:

```
_progres
```

- The server must be running before you issue this command with the *`db-name`* parameter. If the server is not running, OpenEdge displays the following error message:

```
There is no server for database db-name (1423)
```

PRO command

Starts a single-user mode character OpenEdge client session.

Syntax

```
pro [db-name] [parameters]
```

Parameters

db-name

Specifies the database you want to start.

parameters

Specifies the parameters you want to use. You can use any single-user startup parameter with the `PRO` command.

Notes

The `PRO` command runs the following executable:

```
_progres -1
```

PROWIN32 command (Windows only)

Starts a graphical OpenEdge session.

Syntax

```
prowin32 [db-name] [parameters]
```

Parameters

db-name

Specifies the database you are using.

parameters

Specifies the session or connection parameters you want to use.

Notes

See *OpenEdge Deployment: Managing ABL Applications* for more information about starting OpenEdge sessions and connecting to databases.

3

Startup Parameter Usage Categories

This chapter provides an overview of the OpenEdge startup parameters.

This chapter is a a quick reference to locate the appropriate startup parameters to use. Note that some parameters are listed in multiple categories. For detailed descriptions of the startup parameters, see Startup Parameter Descriptions on page 57.

Note: You can specify one or more startup parameters on the command line or incorporate them into a `.BAT` file or a UNIX script file. You can also specify startup parameters using an OpenEdge startup parameter file. For more information on startup parameter files, see Introduction on page 21.

For details, see the following topics:

- Client Session parameters (CS)
- Client Connection parameters (CC)
- OpenEdge Database Server parameters (DBS)
- DataServer parameters (DS)
- OpenEdge SQL Broker parameters (SQL)

Client Session parameters (CS)

Use the Client Session parameters when starting a session for an ABL (Advanced Business Language) client or an AppServer™ (as client). The parameters are organized alphabetically by syntax in the following subcategories:

- Client performance parameters on page 36
- Client product-type parameters on page 37
- ABL parameters on page 38
- Client internationalization parameters on page 42
- Session statistics parameters on page 43
- Windows session parameters on page 43

Client performance parameters

Use the parameters listed in the following table to control client performance.

Syntax	Parameter	Purpose
`-asyncqueuesize` *n*	Async Queue Size	Specifies the total number of bytes allowed for asynchronous request buffers allocated for a client
`-Bt` *n*	Number of Buffers for Temporary Tables	Specifies the size of the buffer pool for blocks or records from the temporary table
`-D` *n*	Directory Size	Changes the number (soft limit) of compiled procedure directory entries
`-fc` *num-entries*	Schema Field Cache Size	Changes the number (soft limit) of entries in the schema field cache
`-hardlimit`	Hardlimit	Forces OpenEdge to adhere to set limits
`-l` *n*	Local Buffer Size	Specifies the size (soft limit) of the local record buffer
`-mc`	Network Message Compression	Compresses messages between an OpenEdge client and the OpenEdge AppServer
`-mmax` *n*	Maximum Memory	Sets the initial amount (soft limit) of memory allocated for r-code segments
`-Mr` *n*	Record Buffer Size	Specifies the standard record buffer size in bytes
`-noautoreslist`	No Auto Result-list	Avoids building result-lists for static non-scrolling queries
`-plm`	`PROLIB` Memory	Allocate a 512-byte cache for the OpenEdge library directory

Syntax	Parameter	Purpose
`-pls`	`PROLIB` Swap	Stores r-code files locally in temporary sort files
`-q`	Quick Request	Directs the ABL Virtual Machine (AVM) to search `PROPATH` directories only on the first use of a procedure
`-s` *`n`*	Stack Size	Specifies the stack size
`-stsh` *`n`*	Stash Area	Specifies the number of 1KB blocks to allocate in the stash area
`-T` *`dirname`*	Temporary Directory	Specifies the directory for temporary files
`-t`	Save Temp Files (UNIX only)	Makes temporary file entries visible
`-TB` *`blocksize`*	Speed Sort	Specifies the block size in kilobytes to allocate when sorting records for reports or rebuilding indexes
`-TM` *`n`*	Merge Number	Specifies the number of blocks or streams simultaneously merged during a sort
`-tmpbsize` *`n`*	Temporary Table Database Block Size	Specifies the temporary table database block size

Client product-type parameters

Use the parameters listed in the following table to control the nature of a client session.

Syntax	Parameter	Purpose
`-noevalprops`	No Evaluation of Properties	Disables evaluation of ABL properties by the OpenEdge Debugger
`-rg`	Run ABL Client	Limits access or functionality of the session to the ABL client
`-rq`	Run query Client	Limits access or functionality of the session to the Query client
`-rr`	Run Run-time Client	Limits access or functionality of the session to the run-time client
`-rx`	Encrypted Compiler Mode	Enables a query or run-time OpenEdge version to compile encrypted source code and access the Data Dictionary to manage schema (for example: for security management and to dump or load the `.df` file)

ABL parameters

Use the parameters listed in the following table to modify ABL behavior.

Syntax	Parameter	Purpose
`-assemblies` *`assemblies-path`*	Assemblies	Specifies the path to the directory containing the Assembly References File (`assemblies.xml`) and any third-party assemblies
`-b`	Batch	Initiates a batch process with no terminal interaction
`-baseADE`	ADE R-code Location	Specifies the ADE r-code directory to the client so the client can add this directory and all procedure libraries contained in the directory to the `PROPATH`
`-browcoltype`	Browse Column Type	Overrides the `TYPE` attribute for browse columns, and always return "`BROWSE-COLUMN`" regardless of the actual column type
`-checkwidth` *`n`*	Check Width	Specifies whether ABL compares `CHARACTER`, `DECIMAL`, and `RAW` field data against the metaschema `_width` field value before updating a database record
`-clearlog`	Clear Log File	Deletes any file that matches the name of the specified client log file identified by the `-clientlog` parameter
`-clientlog`	Client Logging	Allows an application to automatically write all error and warning messages to the specified log file
`-cwl` *`filename`*	Compile Warning List	Flags instances of statements listed in *`filename`* during compilation
`-combofont`	Combo-box Font	Changes the default font of `DECIMAL` values displayed in combo-boxes with `LIST-ITEMS` or `LIST-ITEM-PAIRS`
`-D` *`n`*	Directory Size	Changes the number (soft limit) of compiled procedure directory entries
`-debug`	Debugger	Starts a session by running the Debugger in stand-alone mode
`-debugalert`	Debug Alert	Makes ABL stack trace and .NET stack trace information available during a session, either in an Alert dialog box or a log file
`-debugReady` { `0` | *`port-number`* }	Enable Attachable Debugging	Enables an OpenEdge process (that is, an ABL client or a single WebSpeed agent) to attach to the attachable Debugger

Syntax	Parameter	Purpose
`-defaultscrolling`	Default Scrolling	Makes scrolling the default behavior for all statically defined queries
`-dictexps`	Dictionary Expressions	Directs ABL to use dictionary and help validation for all fields in all frames when compiling
`-disabledeltrig`	Disable Delete Trigger	Disables the delete trigger when ALLOW-REPLICATION is enabled for the DISABLE TRIGGERS statement or the DISABLE-LOAD-TRIGGER() buffer method
`-errorstack`	Error Stack	Allows error objects to save the ABL call stack in the `CallStack` property of an error object at the time the AVM generates the error object
`-expandbrow`	Expand Browse	Sets the `EXPANDABLE` and `FIT-LAST-COLUMN` option to on for all browsers created in the current session
`-filterocxevents`	Filtering Asynchronous COM Events	Controls the conditions under which you want asynchronous COM events handled
`-fldisable`	Field List Disable	Directs ABL to ignore field lists in the r-code and fetch complete records
`-h` *`n`*	Number of Databases	Specifies the maximum number of databases that can be connected
`-icfparam`	Dynamics Parameter	Specifies a character string that can be accessed from ABL procedures within the Dynamics framework
`-inp` *`n`*	Input Characters	Expands the available buffer space for a single statement
`-k` *`filename`*	Keyword Forget List	Disables the keywords listed in *`filename`*
`-literalquestion`	Literal Question	Changes the default value of the `LITERAL-QUESTION` attribute to `TRUE`
`-lkwtmo` *`seconds`*	Lock Timeout	Specifies a different "lock wait timeout"
`-logentrytypes` *`string`*	Log Entry Types	Specifies one or more types of log entries to write to the log file
`-logginglevel` *`n`*	Logging Level	Specifies the level at which log entries are written to the log file
`-logthreshold` *`n`*	Log Threshold	Specifies the file size threshold of log files
`-nb` *`n`*	Nested Blocks	Sets the maximum number (soft limit) of nested procedure blocks
`-NL`	No Lock	Makes all record retrieval statements default to `NO-LOCK` record access

Syntax	Parameter	Purpose
`-noautoreslist`	No Auto Result-list	Avoids building result-lists for static non-scrolling queries
`-nochkttnames`	No Check Temp-table Names	Suppresses the limited checking of temp-table column names when passing temp-tables as parameters to procedures
`-nogc`	No Garbage Collection	Disables automatic garbage collection of class-based objects
`-noinactiveidx`	No Inactive Indexes	Excludes inactive indexes from the compilation of `WHERE` clauses, as well as `LIKE` clauses in `DEFINE TEMP-TABLE` statements
`-noincrwarn`	No Auto-Increment Warnings	Suppresses messages (`5407`) through (`5410`)
`-noint64`	No INT64	Changes the data type of long integer constants to be DECIMAL as opposed to INT64. Does not affect references to variables, fields, or expressions if they are declared as INT64.
`-noroutineinwhere`	No ROUTINE in WHERE Clause	Restricts the invocation of the following from `WHERE` clauses: • Class method • User-defined function • Class property, which implements the `GET` method using ABL statements
`-noudfinwhere`	No UDF in WHERE Clause	Restricts the invocation of user-defined functions and methods from `WHERE` clauses
`-numlogfiles` *`n`*	Number of Log Files to Keep	Specifies the number of rolled over log files to keep on disk at any one time
`-o` *`printername`*	Printer	Identifies the printer to use when processing `OUTPUT TO PRINTER` statements
`-ojmode`	Outer-join Mode	Specifies the mode in which mixed inner and left outer joins, in queries of three or more joined tables, are processed
`-p` *`filename`*	Startup Procedure	Specifies a procedure that executes when OpenEdge starts
`-param` *`string`*	Parameter	Supplies information in a character string or comma-separated list of files to open in Procedure Editor buffers
`-preloadCLR`	Preload CLR	Loads the .NET Common Language Runtime (CLR) into the ABL session at startup
`-proxyhost`	Proxy Host	Specifies the name of the host or the IP address of the host at which the HTTP-based proxy server is located

Syntax	Parameter	Purpose
`-proxyPassword`	Proxy Password	Authenticates the AppServer client to the HTTP-based proxy server
`-proxyport`	Proxy Port	Specifies the port on which the HTTP-based proxy server is listening
`-proxyUserid`	Proxy Userid	Authenticates an AppServer client to the HTTP-based proxy server
`-rand` *n*	Alternate Random Number Generator	Specifies a different random number generator. Use the value 1 for the old generator or the value 2 for a new one
`-rereadnolock`	Reread Nolock	Directs the AVM to re-read a record from the database, if the record is already in a buffer
`-reusableObjects`	Re-usable Objects Cache	Specifies the number of deleted class objects that the AVM stores for later re-initialization.
`-scrvalmode`	Screen-value Mode	Specifies the mode in which the `SCREEN-VALUE` attribute indicates that the selected item in a combo-box item list is empty
`-showppuerr`	Show PPU Error	Starting with Release 10.1B, this startup parameter has no effect; it is supported only for backward compatibility. In earlier releases, it restores the warning for message `4132` ("Invalid character unit value `<N>`. Changed to `320.`")
`-swl`	Suppress Warnings list	Allows ABL programmer to suppress any list of warnings using `SESSION` attribute or startup parameter.
`-ttmarshal` *n*	Temp-table Schema Marshall	Specifies the amount of schema information to marshal for temp-table parameters during an OpenEdge client session
`-tok` *n*	Token	Specifies the maximum number of tokens allowed in an ABL statement
`-undothrow` *n*	UNDO, THROW Error Handling	Changes default error handling to be UNDO, THROW. Use the value 1 to change the default for routine-level blocks. Use the value 2 to change the default for routine-level and other blocks that have default error handling.
`-usewidgetid`	Use Widget ID	Enables application-defined widget IDs for ABL widgets in OpenEdge GUI applications
`-v6colon`	Version 6 Colon	Directs the AVM to use colon alignment of unlabeled fields (as in Version 6)
`-v6q`	Version 6 Query	Directs the OpenEdge database server to use only a single index to resolve `FOR EACH` statements (as in Version 6)

Client internationalization parameters

Use the parameters listed in the following table to control the format in which data appears.

Syntax	Parameter	Purpose
`-checkdbe`	Check Double-byte Enabled	Displays compile-time warnings on ABL statements and functions that may impact the handling of double-byte or extended-character data
`-convmap` *`filename`*	Conversion Map	Identifies the conversion map file
`-cpcase` *`tablename`*	Case Table	Identifies the case table that establishes case rules for the code page
`-cpcoll` *`tablename`*	Collation Table	Identifies a collation table to use with the code page
`-cpinternal` *`codepage`*	Internal Code Page	Identifies the code page that the AVM uses in memory
`-cplog` *`codepage`*	Log File Code Page	Identifies the code page used for writing messages to the log file
`-cpprint` *`codepage`*	Print Code Page	Identifies the code page used for printer output
`-cprcodein` *`codepage`*	R-code in Code Page	Identifies the code page used for reading r-code text segments
`-cprcodeout` *`codepage`*	R-code Out Code Page	Identifies the code page used for writing r-code text segments
`-cpstream` *`codepage`*	Stream Code Page	Identifies the code page used for stream I/O
`-cpterm` *`codepage`*	Terminal Code Page	Identifies the code page for character terminals
`-d` *`dateform`*	Date Format	Specifies the format for displaying dates, using the three-character string of d, m, y in any order
`-E`	European Numeric Format	Displays in European numeric format
`-isnoconv`	Initial Value Segment No Convert	Disables a code page conversion that was not provided in earlier releases of OpenEdge
`-lng` *`languagename`*	Language	Identifies the language with a character string, in quotes
`-numdec` *`numeric-value`*	Fractional Separator	Changes the character that represents a number's decimal point in formatted text
`-numsep` *`numeric-value`*	Thousands Separator	Changes the character that represents a number's thousands separator in formatted text

Syntax	Parameter	Purpose
`-yr4def`	Four Digit Year Default	Outputs a four-digit year from `EXPORT`, `MESSAGE` and `PUT UNFORMATTED` statements that may use a two-digit year
`-yy` *n*	Century Year Offset	Specifies the start of the 100-year period used to define the two-digit `DATE` value

Session statistics parameters

Use the parameters listed in the following table to collect statistics about client session activity and performance.

Syntax	Parameter	Purpose
`-y`	Statistics	Collects procedure access and usage statistics
`-yc`	Statistics with **CTRL+C**	Substitutes **CTRL+C** for executing the `SHOW-STATS` statement
`-yd`	Segment Statistics	Writes segment statistics to the client monitor file (`client.mon` by default)
`-yx`	Statistics with Cross-reference	Collects procedure call statistics and write them to an output file (`proc.mon` by default)
`-ttbasetable`	Base table of the temp-table	Specifies a starting temp-table number in a range of temp-tables for which you want to track access statistics; use with `-tttablerangesize`.
`-ttbaseindex`	Base index of the temp-table	Specifies a range of temp-table indexes for which you want to collect index statistics; use with `-ttindexrangesize`.
`-tttablerangesize`	Table range size of temp-table	Specifies the number of temp-tables for which you want to collect statistics.
`-ttindexrangesize`	Index range size of temp-table	Specifies the number of indexes of temp-tables for which you want to collect statistics.

Windows session parameters

Use the parameters listed in the following table to control a Windows session.

Syntax	Parameter	Purpose
`-basekey` *`name`*	Registry Base Key[1]	Identifies the registry basekey to use for environment information
`-evtlevel` *`value`*	Event Level	Specifies the level of information written to the Windows Event Log
`-ininame` *`name`*	Initialization File[1]	Identifies the initialization registry sub-key or the application's initialization (`.INI`) file
`-Wa -wpp`	Windows Passthrough Printing[1]	Enables Windows Passthrough Printing mode
`-wss`	Windows Single Session	Limits a Windows GUI user to running one client session at a time
`-wy`	Windows Exit - No Dialog	Prevents prompt for confirmation when Windows shuts down while OpenEdge is active

Client Connection parameters (CC)

Use the Client Connection parameters when connecting to an OpenEdge Database Server, AdminServer, AppServer, or a non-OpenEdge DataServer. The parameters are organized alphabetically by syntax. For more information on connecting to a database, see *OpenEdge Deployment: Managing ABL Applications*.

Client database connection parameters

Use the parameters listed in the following table to control how you connect to a specified database.

If you use the Physical Database Name (`-db`) parameter to specify more than one database when you start a session, specify the connection parameters for each database after the database name to which they apply and before the next `-db` parameter. OpenEdge applies database connection parameters only to the previously specified database. For example:

```
mpro [ -db] db1-name[db1-parameters ] -db db2-name [db2-parameters ]
```

Note: OpenEdge implicitly specifies the `-db` parameter for the first database.

You can specify all other parameters anywhere on the command line. If you specify the same parameter more than once, OpenEdge uses the value you provide for the last instance of the parameter.

[1] You can use this parameter only on the command line. It is not supported in a parameter (`.pf`) file.

Syntax	Parameter	Purpose
`-1`	Single-user Mode[2]	Starts OpenEdge in single-user mode
`-B` *n*	Blocks in Database Buffers[2]	Specifies the number of database buffer blocks
`-B2` *n*	Blocks in Alternate Buffer Pool[2]	Specifies the number of buffers in the database Alternate Buffer Pool
`-Bp` *n*	Private Read-only Buffers	Requests a number of private read-only buffers
`-brl`	Bleeding Record Lock	Holds a share lock on a released record used by more than one buffer
`-cache` *filename*	Schema Cache File	Reads the database schema from a local file instead of the database
`-cp` { *com-parms* | *filename* }	Communications Parameter File	Overrides preset connection parameters for auto-connect databases
`-crStatus` *n*	Crash Recovery Status	Controls how often a crash recovery status message is displayed
`-crTXDisplay`	Crash Recovery Transaction Display	Displays the transaction table during crash recovery.
`-cs` *n*	Cursor Size	Changes the maximum number of index levels
`-ct` *n*	Connection Retry Attempts	Defines the number of retries when connecting to a database
`-db` *physical-dbname*	Physical Database Name	Identifies one or more databases to connect to at startup
`-DataService` *data-service*	DataService	Connects through a NameServer to an ODBC or ORACLE DataServer
`-directio`	Direct I/O[2], [3]	Opens all files in unbuffered mode
`-dt` *db-type*	Database Type	Identifies the database type
`-F`	Force Access	Allows access to a damaged database. Call Technical Support before using `-F`
`-H` *host-name*	Host Name	Identifies a remote host

[2] Single-user database connections only; otherwise, use only for server startup.

[3] UNIX, Windows only.

Syntax	Parameter	Purpose
-hash *n*	Hash Table Entries[2]	Specifies the number of hash table entries for the buffer pool
-i	No Crash Protection[2]	Runs OpenEdge without database integrity or recovery
-ipver	Internet Protocol	Specifies the Internet Protocol version
-is	AS/400 Ignore Stamp	Stops a client from comparing time stamps when connection to an AS/400 database
-ld *logical-dbname*	Logical Database Name	Assigns the logical database name
-Mf *n*	Delayed BI File Write[2, 3]	Delays writing the last before-image (BI) file records
-Mm *n*	Message Buffer Size	Specifies the message buffer size for network client/server protocols
-nohostverify	No Host Verify	Turns off host verification for a Secure Sockets Layer (SSL) connection
-nosessionreuse	No Session Reuse	Avoids the reuse of a Secure Sockets Layer (SSL) session ID
-P *password*	Password[4]	Specifies a user password
-Passphrase	Prompt for Passphrase[2]	When connecting to a database enabled for Transparent Data Encryption, specifies that the user must be prompted for the key store passphrase
-populate	Fast Schema Change	Turns off Fast schema change
-r	Non-reliable I/O[2]	Enables buffered I/O to the before-image file
-requireusername	Require Username	Requires users starting servers of the AdminServer to provide a valid username and password
-RO	Read-only Media	Opens a database for read-only access
-S { *servicename* | ***port-number*** }	Service Name	Identifies the service and port number to use on the host machine when connecting to a broker process
-Sn	AS/400 Server Program Name	Specifies that the Server program on an AS/400 start with an OpenEdge connection request (SNA only)
-ssl	SSL-based Connection	Specifies a Secure Socket Layer connection

[4] Different meaning for non-OpenEdge databases.

Syntax	Parameter	Purpose
`-tstamp`	Time Stamp	Uses time stamp information rather than cyclic redundancy (CRC) to enforce consistency between r-code and a schema

`-trig`
{ *`dir-name`* | *`lib-name`* }
	Trigger Location	Identifies the directory or library containing the ABL triggers for a database

`-U` *`userid`*	User ID[4]	Specifies the user ID

`jdbc:datadirect:openedge://<ip>:<port>;`
`databaseName=<dbname>;`
`truncateTooLarge=<on/off>;`
	truncateTooLarge	With its value set to *`ON`*, the parameter allows truncating data exceeding the column size. If the value is set to *`OFF`*, no data is truncated and an error is displayed. The default value for the
		`truncateTooLarge`
		parameter is *`OFF`*. Not specifying the parameter is equivalent to setting its value to *`OFF`*.

OpenEdge Database Server parameters (DBS)

The OpenEdge Database Server parameters are organized alphabetically by syntax in the following subcategories:

- Database server performance parameters on page 47
- Database server-type parameters on page 51
- Database server internationalization parameters on page 52
- Database server statistics collection parameters on page 52
- Database server consistency check parameters on page 53
- Database server network parameters on page 53

Database server performance parameters

Use the parameters listed in the following table to optimize server performance.

Syntax	Parameter	Purpose
`-aiarcdir` *`dirlist`*	After-image File Management Archive Directory List	Specifies the directories where the AI File Management utility writes the archived after-image files

Syntax	Parameter	Purpose
`-aiarcdircreate`	After-image File Management Archive Directory Create	Directs the AI File Management utility to create the specified directories
`-aiarcinterval` *n*	After-image File Management Archive Interval	Specifies on-demand mode archiving or the extent switch interval for timed-mode archiving
`-aibufs` *n*	After-image Buffers	Specifies the number of after-image buffers when running AIW
`-aistall`	After-image Stall	Suspends database activity when an empty after-image (AI) file is unavailable
`-B` *n*	Blocks in Database Buffers	Specifies the number of blocks in the database buffers
`-B2` *n*	Blocks in Alternate Buffer Pool	Specifies the number of buffers in the database Alternate Buffer Pool
`-Bpmax`	Maximum Private Buffers per Connection	Controls the number of private buffers a connection can request through the Private Read-only Buffers (`-Bp`) parameter
`-bibufs` *n*	Before-image Buffers	Specifies the number of before-image buffers when running BIW
`-bistall`	Threshold Stall	Sends a message to the log file when the recovery log threshold is reached. Use with `-bithold`.
`-bithold` *n*	Recovery Log Threshold	Specifies the maximum size of the recovery log files
`-cluster` *qualifier*	Cluster Mode	Starts up a cluster-enabled database
`-directio`	Direct I/O (UNIX, Windows only)	Opens all files in unbuffered mode
`-G` *n*	Before-image Cluster Age	Specifies the number of seconds before OpenEdge reuses a before-image cluster
`-groupdelay` *n*	Group Delay	Specifies the number of milliseconds a transaction waits before committing
`-hash`	Hash Table Entries	Specifies the number of hash table entries for the buffer pool
`-i`	No Crash Protection	Runs OpenEdge without using database integrity or recovery
`-LGovernor` *n*	Lock Governor	Specifies the percentage of the lock table that a tenant can occupy

Syntax	Parameter	Purpose
`-L` *n*	Lock Table Entries	Specifies the number of entries in the record locking table
`-lkhash` *n*	Lock Table Hash Size	Specifies the size of the hash table controlling access to the lock table
`-lkrela`	Lock Release	Uses the original lock release mechanism installed from previous OpenEdge versions
`-lruskips`	LRU force skips	Specify the number of times a buffer in the buffer pool is accessed before it is placed on the Most Recently Used (MRU) end of the LRU chain
`-lru2skips`	LRU alternate buffer pool force skips	Specify the number of times a buffer in the alternate buffer pool is accessed before it is placed on the Most Recently Used (MRU) end of the LRU chain
`-maxAreas` *n*	Maximum Area Number	Specifies the maximum area number available
`-maxxids` *n*	Maximum JTA Transactions	Controls the number of JTA transactions simultaneously allowed
`-Mf` *n*	Delayed BI File Write (UNIX, Windows only)	Delays writing the last before-image (BI) file records
`-Mm` *n*	Message Buffer Size	Specifies the message buffer size for network client/server protocols
`-Mpte`	VLM Page Table Entry Optimization (digital UNIX only)	Allocates shared memory in multiples of 8MB for VLM64 support
`-mux` *n*	User MUX latches	Specifies the granularity of access to large database resources in shared memory
`-Mxs` *n*	Shared-memory Overflow Size (UNIX, Windows only)	Replaces the default value of the shared-memory overflow area
`-n` *n*	Number of Users	Specifies the maximum number of users connected to the database
`-napinc` *n*	Naptime Increment	Specifies the nap time increment
`-napmax` *n*	Nap Maximum	Specifies the maximum time (in milliseconds) to sleep (nap) after the `-spin` value is exhausted due to failure to acquire a latch
`-napstep` *n*	Naptime Steps	Specifies the steps between the nap increments
`-nGovernor` *n*	Login Governor	Specifies the maximum number of users that may login for a tenant

Syntax	Parameter	Purpose
`-Nmsgwait` *n*	Message wait	Specifies the number of seconds a server waits for a remote network message
`-omsize` *n*	Storage Object Cache Size	Specifies the size of the object cache for all database objects
`-Passphrase`	Prompt for Passphrase	When connecting to a database enabled for Transparent Data Encryption, specifies that the user must be prompted for the key store passphrase
`-PendConnTime` *n*	Pending Connection Time	Specifies the amount of time a client has to connect to a server before the broker clears the client's reservation
`-pica`	Database Service Communication Area Size	For OpenEdge Replication, specifies the size of the database service communications area in KB
`-pinshm`	Pin Shared Memory	Prevents the database engine from swapping shared memory contents to disk
`-prefetchDelay`	Prefetch delay	Enable a delay when sending the first network message for queries with prefetch capabilities
`-prefetchFactor` *n*	Prefetch factor	Specify a percentage of a network message required to contain prefetched data
`-prefetchNumRecs` *n*	Prefetch num recs	Specify the number of prefetch records placed in a network message
`-prefetchPriority` *n*	Prefetch priority	Specify the number of prefetch records to add to an in-process query without polling
`-schlockwq`	Schema Lock Wait Queue	Alters the schema locking algorithm so that the schema lock scheduler forces a shared schema lock request to wait when an exclusive schema lock request is queued
`-semsets` *n*	Semaphore Sets (UNIX only)	Changes the number of semaphore sets available to a broker
`-shmsegsize` *n*	Shared Memory Segment Size	Specifies the maximum shared memory segment size
`-spin` *n*	Spin Lock Retries (UNIX, Windows only)	Specifies the number of times a process tries to acquire a latch before pausing
`-SQLCursors` *value*	SQL Open Cursors	Specifies the maximum number of cursors open at any one time
`-SQLStack` *n*	SQL Stack Size	Specifies the size, in KB, of the SQL Stack
`-SQLStmtCache` *num-entries*	SQL Statement Cache Size	Specifies the number of statements allowed in the SQL statement cache

Syntax	Parameter	Purpose
`-SQLTempBuff` *value*	SQL Sorting Memory	Defines the size of the temporary table buffer in memory
`-SQLTempDisk` *value*	SQL Sorting on Disk	Defines the size of the temporary table for backup storage
`-SQLTempPgSize` *value*	SQL Temp Table Data Page	Defines the size of the temporary table data page
`-TXERetryLimit` *n*	Transaction End (TXE) Lock Retry Limit	Sets the number times a connection will re-try to obtain TXE lock after a 1ms nap before getting queued.
`SQLTruncateTooLarge` *value*	Authorized Data Truncation	Can be set to *ON* to truncate data exceeding the column size. If the parameter is set to *OFF*, no data is truncated and SQL displays an error. SQL also displays an error if the SQLTruncateTooLarge parameter is not specified at all in case of data exceeding the column size. The default value for the SQLTruncateTooLarge parameter is *OFF*.
`-SQLWidthUpdate` *value*	Autonomous Schema Update	Helps resolve the SQL width problem by updating the SQL column width in schema automatically when Authorized Data Truncation(ADT) occurs for data in a column. To use this feature, provide the value of the -SQLWidthUpdate parameter as *ON* during startup. Once the value of the parameter is set, it is remembered for the lifetime of the server and for all connections. If the value of the -SQLWidthUpdate parameter is set to *ON* during startup, the Authorized Data Truncation(ADT) parameter is also enabled irrespective of whether ADT was enabled during startup unless specified otherwise using connection URL. **Note:** This parameter can be provided only during server startup and not in the client connection URL.

Database server-type parameters

Use the parameters listed in the following table to start a particular type of server.

Syntax	Parameter	Purpose
`-m1`	Auto Server	Starts an auto server; used internally by the database broker
`-m2`	Manual Server (UNIX, Windows only)	Manually starts a remote user server after you start a broker

Syntax	Parameter	Purpose
`-m3`	Secondary Login Broker (UNIX, Windows only)	Starts a secondary broker
`-ServerType [ 4GL | SQL | Both ]`	Type of Server to Start	Specifies whether to start an OpenEdge server, SQL server, or both

Database server internationalization parameters

Use the parameters listed in the following table to control the format in which data appears.

Syntax	Parameter	Purpose
`-convmap` *filename*	Conversion Map	Identifies the conversion map file
`-cpcase` *tablename*	Case Table	Identifies the case table that establishes case rules for the code page
`-cpcoll` *tablename*	Collation Table	Identifies a collation table to use with the code page
`-cpinternal` *code-page*	Internal Code Page	Identifies the code page that the AVM uses in memory
`-cplog` *code-page*	Log File Code Page	Identifies the code page used for writing messages to the log file
`-cpprint` *code-page*	Print Code Page	Identifies the code page used for printer output
`-cprcodein` *code-page*	R-code in Code Page	Identifies the code page used for reading r-code text segments
`-cpstream` *code-page*	Stream Code Page	Identifies the code page used for stream I/O
`-cpterm` *code-page*	Terminal Code Page	Identifies the code page for character terminals

Database server statistics collection parameters

Use the parameters listed in the following table to collect statistics for table and index access.

Syntax	Parameter	Purpose
`-baseindex` *n*	Base Index	Specifies a range of indexes for which you want to collect statistics; use with `-indexrangesize`
`-basetable` *n*	Base Table	Specifies a starting table number in a range of tables for which you want to track access statistics; use with `-tablerangesize`

Syntax	Parameter	Purpose
`-indexrangesize` *n*	Index Range Size	Specifies the number of indexes to track for access statistics
`-tablerangesize` *n*	Table Range Size	Specifies the number of tables for which you want to collect statistics

Database server consistency check parameters

Use the parameters listed in the following table to enable consistency checking.

Syntax	Parameter	Purpose
`-AreaCheck` *area-name*	Area consistency check	Area consistency check enables consistency checking for all record and index blocks in the specified area
`-DbCheck`	Database consistency check	Database consistency check enables consistency checking for all record and index blocks in the entire database
`-MemCheck`	Memory consistency check	Memory consistency check enable memory overwrite checking for the buffer pool
`-IndexCheck` *index-name*	Index consistency check	Index consistency check enables consistency checking for all index blocks in the specified index
`-TableCheck` *table-name*	Table consistency check	Table consistency check enables consistency checking for all record blocks of the specified table (excluding BLOBS)

Database server network parameters

Use the parameters listed in the following table to supply OpenEdge with network information.

Syntax	Parameter	Purpose

`-adminport`
{ *service-name* | *port* }
	AdminServer Port	Connects a servergroup and an AdminServer

`-classpath` *pathname*	SQL Server Java™ Classpath	Identifies the Java classpath to use when starting an SQL server
`-H` *host-name*	Host Name	Specifies a remote host

Syntax	Parameter	Purpose
`-keyalias` *`key-alias-name`*	Key Alias	Specifies the alias name of a Secure Sockets Layer (SSL) private key/digital certificate key-store
`-keyaliaspasswd` *`key-alias-password`*	Key Alias Password	Specifies the encrypted Secure Sockets Layer (SSL) key alias password
`-Ma` *`n`*	Maximum Clients per Server[5]	Specifies the maximum number of remote users per database server
`-maxport` *`n`*	Maximum Dynamic Server[5]	Specifies the highest accessible port number in a specified range
`-Mi` *`n`*	Minimum Clients per Server[5]	Specifies the number of remote users on a server before a broker starts another server
`-minport` *`n`*	Minimum Dynamic Server[5]	Specifies the lowest accessible port number in a specified range
`-Mn` *`n`*	Maximum Servers[5]	Specifies the maximum number of remote client servers that a broker can start
`-Mp` *`n`*	Servers per Protocol[5]	Specifies the maximum number of servers to serve remote users for a protocol
`-Mpb` *`n`*	Maximum Servers per Broker[5]	Specifies the maximum number of servers that multiple brokers can start to serve remote users for a protocol
`-nosessioncache`	No Session Cache	Disables the Secure Sockets Layer (SSL) session caching
`-properties` *`filename`*	Configuration Properties File	Identifies the properties file an AdminServer uses when starting a database server or servergroup
`-S` { *`service-name`* | *`port-number`* }	Service Name	Specifies the service or port number to be used by a broker process
`-servergroup` *`name`*	Server Group	Identifies a logical collection of server processes to start
`-sessiontimeout` *`n`*	Session Timeout	Specifies the length of time a SSL session is held in SSL session cache
`-ssl`	SSL-based Connection	Specifies a Secure Socket Layer (SSL) to all database and client connections

[5] UNIX, Windows only.

DataServer parameters (DS)

Use the parameters listed in the following table when starting an ODBC, ORACLE, or MS SQL Server DataServer. The parameters are organized alphabetically by syntax.

Syntax	Parameter	Purpose
`-c` *n*	Index Cursors	Specifies the number of index place holders or *cursors*
`-cpinternal` *code-page*	Internal Code Page	Identifies the code page that the AVM uses in memory
`-cplog` *code-page*	Log File Code Page	Identifies the code page used for writing messages to the log file
`-cpprint` *code-page*	Print Code Page	Identifies the code page used for printer output
`-cprcodein` *code-page*	R-code in Code Page	Identifies the code page used for reading r-code text segments
`-cpstream` *code-page*	Stream Code Page	Identifies the code page used for stream I/O
`-cpterm` *code-page*	Terminal Code Page	Identifies the code page for character terminals
`-dslog` *filename*	DataServer Logging	Allow an application to automatically write server error and warning messages, and other server log messages, to the specified log file (MS SQL Server only)
`-dsmaxport` *n*	Maximum Dynamic DataServer	Specifies the highest port number in a specified range of port numbers accessible to a client
`-dsminport` *n*	Minimum Dynamic DataServer	Specifies the lowest port number in a specified range of port numbers accessible to a client
`-Dsrv` *keyword* (, *value*) [, *keyword2* (, *value2*) ...]	DataServer	Identifies keywords as parameters for the ODBC, ORACLE, or MS SQL Server DataServer
`-dt` *db-type*	Database Type	Identifies the database type
`-H` *host-name*	Host Name	Specifies a remote host

Syntax	Parameter	Purpose
`-is`	AS/400 Ignore Stamp	Stops a client from comparing time stamps when connecting to an AS/400 database
`-noindexhint`	Index Hint	Stops an ORACLE DataServer from providing index hints to the ORACLE DBMS
`-nojoinbysqldb`	Server Join	Makes a client evaluate and perform queries that have joins
`-noSQLbyserver`	SELECT Pass Through Disable	Turns off SELECT pass Through for OBDC and ORACLE DataServers
`-Sn`	AS/400 Server Program Name	Starts a Server program on an AS/400 with a OpenEdge connection request (SNA only)

OpenEdge SQL Broker parameters (SQL)

Use the parameters listed in the following table to control spawned OpenEdge SQL processes when connecting to a OpenEdge SQL server. The parameters are organized alphabetically by syntax.

Syntax	Parameter	Purpose
`-classpath`	SQL Server Java Classpath	Identifies the Java classpath to use when starting an SQL server
`-ServerType`	Type of Server to Start	Limits the type of server the broker can start
`-SQLCursors`	SQL Open Cursors	Defines the maximum number of cursors open at any one time
`-SQLQuotedRowid`	SQL rowid identifier	Specify the interpretation of the quoted string "rowid"
`-SQLStack`	SQL Stack Size	Changes the size of the SQL stack
`-SQLStmtCache`	SQL Statement Cache	Sets the number of statements that can be stored in the SQL cache
`-SQLTempBuff`	SQL Sorting Memory	Defines the size in K-bytes of the temporary table buffers in main memory cache

4

Startup Parameter Descriptions

This chapter provides a detailed reference description for each of the OpenEdge startup parameters, organized in alphabetical order by syntax.

Note: Unless otherwise indicated, startup parameter syntax is the same for both UNIX and Windows.

For details, see the following topics:

- Blocks in Alternate Buffer Pool (-B2)
- Batch (-b)
- ADE R-code Location (-baseADE)
- Base Index (-baseindex)
- Registry Base Key (-basekey)
- Base Table (-basetable)
- Before-image Buffers (-bibufs)
- Threshold Stall (-bistall)
- Recovery Log Threshold (-bithold)
- Private Read-only Buffers (-Bp)
- Maximum Private Buffers per Connection (-Bpmax)
- Bleeding Record Lock (-brl)
- Browse Column Type (-browcoltype)
- Number of Buffers for Temporary Tables (-Bt)
- Index Cursors (-c)
- Schema Cache File (-cache)
- Check Double-byte Enabled (-checkdbe)
- Check Width (-checkwidth)
- SQL Server Java Classpath (-classpath)
- Clear Log (-clearlog)
- Client Logging (-clientlog)
- Cluster Mode (-cluster)
- Combo-box Font (-combofont)
- Conversion Map (-convmap)
- Communications Parameter File (-cp)
- Case Table (-cpcase)
- Collation Table (-cpcoll)
- Internal Code Page (-cpinternal)
- Log File Code Page (-cplog)
- Print Code Page (-cpprint)
- R-code in Code Page (-cprcodein)
- R-code Out Code Page (-cprcodeout)
- Stream Code Page (-cpstream)

- Terminal Code Page (-cpterm)
- Crash Recovery Status (-crStatus)
- Crash Recovery Transaction Display (-crTXDisplay)
- Cursor Size (-cs)
- Connection Retry Attempts (-ct)
- Compile Warning List (-cwl)
- Directory Size (-D)
- Date Format (-d)
- DataService (-DataService)
- Physical Database Name (-db)
- Database consistency check (-DbCheck)
- Debugger (-debug)
- Debug Alert (-debugalert)
- Enable Attachable Debugging (-debugReady)
- Default scrolling (-defaultscrolling)
- Disable Delete Trigger (-disabledeltrig)
- Dictionary Expressions (-dictexps)
- Direct I/O (-directio)
- DataServer Logging (-dslog)
- Maximum Dynamic DataServer Port (-dsmaxport)
- Minimum Dynamic DataServer (-dsminport)
- DataServer (-Dsrv)
- Database Type (-dt)
- European Numeric Format (-E)
- Enforce Mm (-enforceMm)
- Entity Expansion Limit (-entityExpansionLimit)
- Error Stack (-errorstack)
- Event Level (-evtlevel)
- Expand Browse (-expandbrow)
- Force Access (-F)
- Schema Field Cache Size (-fc)
- Filtering Asynchronous COM Events (-filterocxevents)
- Field List Disable (-fldisable)

- Before-image Truncate Interval (-G)
- Group Delay (-groupdelay)
- Host Name (-H)
- Number of Databases (-h)
- Hardlimit (-hardlimit)
- Hash Table Entries (-hash)
- No Crash Protection (-i)
- Dynamics Parameter (-icfparam)
- Index consistency check (-IndexCheck)
- Index Range Size (-indexrangesize)
- Initialization File (-ininame)
- Input Characters (-inp)
- Internet Protocol (-ipver)
- Initial Value Segment No Convert (-isnoconv)
- Keyword Forget List (-k)
- Key Alias (-keyalias)
- Key Alias Password (-keyaliaspasswd)
- Lock Table Entries (-L)
- Local Buffer Size (-l)
- Logical Database Name (-ld)
- Lock Governor (-LGovernor)
- Literal Question (-literalquestion)
- Lock Table Hash Size (-lkhash)
- Lock release (-lkrela)
- Lock Timeout (-lkwtmo)
- Language (-lng)
- Log Entry Types (-logentrytypes)
- Logging Level (-logginglevel)
- Log Threshold (-logthreshold)
- LRU force skips (-lruskips)
- LRU alternate buffer pool force skips (-lru2skips)
- Auto Server (-m1)
- Manual Server (-m2)

- Secondary Login Broker (-m3)
- Maximum Clients Per Server (-Ma)
- Maximum Area Number (-maxAreas)
- Maximum Dynamic Server (-maxport)
- Maximum JTA Transactions (-maxxids)
- Network Message Compression (-mc)
- Memory consistency check (-MemCheck)
- Delayed BI File Write (-Mf)
- Minimum Clients per Server (-Mi)
- Minimum Dynamic Server (-minport)
- Message Buffer Size (-Mm)
- Maximum Memory (-mmax)
- Maximum Servers (-Mn)
- Servers Per Protocol (-Mp)
- Maximum Servers Per Broker (-Mpb)
- Record Buffer Size (-Mr)
- User MUX Latches (-mux)
- Shared-memory Overflow Size (-Mxs)
- Number of Users (-n)
- Nap Time Increment (-napinc)
- Nap Maximum (-napmax)
- Nap Time Steps (-napstep)
- Nested Blocks (-nb)
- Login Governor (-nGovernor)
- No Lock (-NL)
- Message Wait (-Nmsgwait)
- No Auto Result-list (-noautoreslist)
- No Domain Support With the CAN-DO Function (-nocandodomain)
- No Check Temp-table Names (-nochkttnames)
- No Colons for Side Labels (-nocolon)
- No Evaluation of Properties (-noevalprops)
- No Garbage Collection (-nogc)
- No Host Verify (-nohostverify)

- No Inactive Indexes (-noinactiveidx)
- No Auto-Increment Warnings (-noincrwarn)
- Index Hint (-noindexhint)
- No INT64 (-noint64)
- Server Join (-nojoinbysqldb)
- No routine in WHERE Clause (-noroutineinwhere)
- No Session Cache (-nosessioncache)
- No Session Reuse (-nosessionreuse)
- SELECT Pass Through Disable (-noSQLbyserver)
- No UDF in WHERE Clause (-noudfinwhere)
- Fractional Separator (-numdec)
- Number of Log Files to Keep (-numlogfiles)
- Thousands Separator (-numsep)
- Printer (-o)
- Outer-join Mode (-ojmode)
- Storage Object Cache Size (-omsize)
- Password (-P)
- Startup Procedure (-p)
- Parameter (-param)
- Prompt for Passphrase (-Passphrase)
- Pending Connection Time (-PendConnTime)
- Parameter File (-pf)
- Database Service Communication Area Size (-pica)
- Pin Shared Memory (-pinshm)
- PROLIB Memory (-plm)
- PROLIB Swap (-pls)
- Fast Schema Change (-populate)
- Prefetch delay (-prefetchDelay)
- Prefetch factor (-prefetchFactor)
- Prefetch Num Recs (-prefetchNumRecs)
- Prefetch Priority (-prefetchPriority)
- Preload CLR (-preloadCLR)
- Configuration Properties File (-properties)

- Proxy Host (-proxyhost)
- Proxy Password (-proxyPassword)
- Proxy Port (-proxyport)
- Proxy Userid (-proxyUserid)
- Quick Request (-q)
- Non-reliable I/O (-r)
- Alternate Random Number Generator (-rand)
- Require Username (-requireusername)
- Reread Fields (-rereadfields)
- Reread Nolock (-rereadnolock)
- Re-usable Objects Cache (-reusableObjects)
- Run ABL Client (-rg)
- Read-only Media (-RO)
- Run Query Client (-rq)
- Run Run-time Client (-rr)
- Encrypted Compiler Mode (-rx)
- Service Name (-S)
- Stack Size (-s)
- Schema Lock Wait Queue (-schlockwq)
- Screen-value Mode (-scrvalmode)
- Semaphore Sets (-semsets)
- Server Group (-servergroup)
- Type of Server to Start (-ServerType)
- Session Timeout (-sessiontimeout)
- Shared memory segment size (-shmsegsize)
- Show PPU Error (-showppuerr)
- Spin Lock Retries (-spin)
- SQL Open Cursors (-SQLCursors)
- SQL rowid identifier(-SQLQuotedRowid)
- SQL Stack Size (-SQLStack)
- SQL Statement Cache Size (-SQLStmtCache)
- SQL Sorting Memory (-SQLTempBuff)
- SSL-based Connection (-ssl)

- Strict Entity Resolution (-strictEntityResolution)
- Stash Area (-stsh)
- Temporary Directory (-T)
- Save Temp Files (-t)
- Table consistency check (-TableCheck)
- Table Range Size (-tablerangesize)
- Speed Sort (-TB)
- Time Stamp (-tstamp)
- Temp-table Schema Marshal (-ttmarshal)
- Merge Number (-TM)
- Temporary Table Database Block Size (-tmpbsize)
- Token (-tok)
- Trigger Location (-trig)
- Temp-Table Base Index (-ttbaseindex)
- Temp-Table Base Table (-ttbasetable)
- Temp-table Index Range Size (-ttindexrangesize)
- Temp-Table Table Range Size (-tttablerangesize)
- Transaction End (TXE) Lock Retry Limit (-TXERetryLimit)
- UNDO, THROW Error Handling (-undothrow)
- User ID (-U)
- Use OS Locale (-useOsLocale)
- Use Widget ID (-usewidgetid)
- Version 6 Colon (-v6colon)
- Version 6 Query (-v6q)
- Windows Passthrough Printing (-Wa -wpp)
- Windows Single Session (-wss)
- Windows Exit - No Dialog (-wy)
- Statistics (-y)
- Statistics with CTRL+C (-yc)
- Segment Statistics (-yd)
- Four Digit Year Default (-yr4def)
- Statistics With Cross-reference (-yx)
- Century Year Offset (-yy)

Single-user Mode (-1)

Use Single-user Mode (`-1`) with an OpenEdge executable module to start OpenEdge in single-user mode. In this mode, only one user can access the database.

Operating system and syntax	UNIX / Windows	**-1**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Connection	—	—	—	—

The PRO command on page 33 invokes the OpenEdge executable with `-1` by default, but only for the first database named on the command line. Each additional database must already have a server running in multi-user mode, unless `-1` is added to its parameter list. In the following example, `db1` and `db3` are connected in single-user mode; `db2` is connected in multi-user mode:

```
pro db1 -db db2 -db db3 -1
```

AdminServer Group (-admingroup)

Use AdminServer Group (`-admingroup`) to authorize OpenEdge AdminServer access and operational privileges for one or more user groups. To perform AdminServer functions, a user must use a valid user account that appears in at least one of the user groups.

Operating system and syntax	UNIX / Windows	**-admingroup** ***user-group*** [: ***user-group***]		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Connection	—	—	—	—

user-group

> The name of the group of users. There must be a minimum of one valid group or the AdminServer will not start. If multiple groups are listed, they are separated by a colon. The user starting the AdminServer must be a valid account in one of the groups.

On UNIX, you can specify a user group as any local group name or NIS group name. Subgroups are not supported. In Windows, you can specify a user group as any local group name or Domain group name. Global groups as members of local groups are supported.

If, during the OpenEdge installation process, you accepted the default installation and did not choose to use authorization, you can use this parameter when starting the AdminServer to select authorization for one or more user groups. Otherwise, no group authorization is defined and all users have authorization.

AdminServer Port (-adminport)

Use AdminServer Port (`–adminport`) to establish communication between a servergroup and an AdminServer.

Operating system and syntax	**UNIX / Windows**	**`-adminport` *`service-name port`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	—	—	—	—

`service-name`

The name of the service to be used by the AdminServer.

`port`

The port number the AdminServer uses to communicate with server groups. The default port is 7844.

Use AdminServer Port (`–adminport`) to establish communication between a server group and an AdminServer. The AdminServer uses this parameter internally. The `–adminport` setting must match the `–admin` setting specified when the AdminServer was started.

After-image File Management Archive Directory List (-aiarcdir)

Use After-image File Management Archive Directory List (-`aiarcdir`) to specify the directories where the AI File Management utility writes the archived after-image files.

Operating system and syntax	**UNIX / Windows**	**`-aiarcdir`** ***`dirlist`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	—	—	—	—

`dirlist`

A comma-separated list of directories where archived after-image files are written by the AI File Management Utility. The directories cannot have any embedded spaces in their names.

The directories must exist, unless you also specify After-image File Management Archive Directory Create (-aiarcdircreate) on page 67 to direct the utility to create the directories.

Note: This parameter is useful only when running the AI File Management utility.

For information on using the AI File Management utility, see *OpenEdge Data Management: Database Administration*.

After-image File Management Archive Directory Create (-aiarcdircreate)

Use After-image File Management Archive Directory Create (-`aiarcdircreate`) to direct the AI File Management utility to create the specified directories.

Operating system and syntax	**UNIX / Windows**	**`-aiarcdircreate`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	—	—	—	—

Note: This parameter is useful only when running the AI File Management utility.

For information on using the AI File Management utility, see *OpenEdge Data Management: Database Administration*.

After-image File Management Archive Interval (-aiarcinterval)

Use After-image File Management Archive Interval (-`aiarcinterval`) to specify the extent switch interval for timed-mode archiving. For on-demand archiving, do not specify this parameter.

Operating system and syntax	**UNIX / Windows**	**`-aiarcinterval` *`n`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	86400	120	—	120

`n`

The number of seconds between mandatory extent switches by the AI File Management utility.

If you specify timed-mode, set the elapsed time between forced AI extent switches with a value indicating the number of seconds between the switches. The minimum time is 2 minutes, the maximum is 24 hours. You can modify the interval on a running system by using the RFUTIL utility.

In both timed and on-demand modes, the AI File Management utility archives full AI extents every 5 seconds.

Note: This parameter is useful only when running the AI File Management utility.

For information on using the AI File Management utility, see *OpenEdge Data Management: Database Administration*.

After-image Buffers (-aibufs)

Use After-image Buffers (`-aibufs`) to specify the number of after-image buffers.

Note: Starting in Release 10.1C, the OpenEdge RDBMS expands the range of internal validations used to ensure database consistency in both index and data blocks during forward processing. Validations using PROUTIL can be run online as part of routine health checks. For more information, see *OpenEdge Data Management: Database Administration*.

Operating system and syntax	**UNIX / Windows**	**`-aibufs n`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	—	1	—	20

n

The number of after-image buffers to use.

This parameter is useful only when running the after-image writer (AIW) because the AIW writes the filled After-image Buffers to disk, making the buffers available to other client and server processes. Progress Software Corporation recommends you set After-image Buffers (`-aibufs`) to the same value specified for the Before-image Buffers (-bibufs) on page 78 parameter, or a minimum of 20.

Without the AIW writing the buffers, any gain from increasing the number of buffers is negligible.

After-image Stall (-aistall)

Use After-image Stall (`-aistall`) to suspend database activity if all AI files are filled.

Operating system and syntax	**UNIX / Windows**	**`-aistall`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	—	—	—	—

Wait until the next AI file is backed up and marked empty, then send the following message to the database log file:

```
Can't switch to after-image extent filename it is full. (3775)
Backup ai extent and mark it as empty (3776)
```

This way you can empty an AI file instead of shutting down the database.

When using after-image (AI) files, you monitor the status of the files to ensure that you do not attempt to reuse an unavailable file. If all the AI files are filled and OpenEdge cannot switch to an empty one, by default the database shuts down. See *OpenEdge Data Management: Database Administration* for information on using the after imaging feature.

Area consistency check (-AreaCheck)

Use Area consistency check to enable consistency checking for all record and index blocks in the specified area.

Operating system and syntax	**UNIX / Windows**	**`-AreaCheck` *`areaname`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Connection, Database Server	—	—	—	—

`areaname`

Specifies the name of the area for consistency checking.

When enabled, this option applies consistency checks to all index blocks and record blocks for record and index write operations. You can only specify one area with `-AreaCheck`. To check multiple areas, you must apply consistency checking to the entire database with Database consistency check (-DbCheck) on page 103.

Area consistency check validates a block is still physically correct after an operation has been performed. For example, after an index key is inserted into an index block, the consistency check validates that the block still laid out correctly.

Area consistency checking can be enabled or disabled while your database is online with PROMON. See *OpenEdge Data Management: Database Administration* for more information.

Area consistency check can be enabled for a single user client or for any offline utility. If the utility is read-only, such as DBANALYS, the `-AreaCheck` parameter is ignored. Online utilities determine whether or not to perform area consistency checking during execution based on the use of `-AreaCheck` at broker startup or by the enablement/disablement of the database consistency checking in PROMON.

Assemblies (-assemblies)

Use Assemblies (`-assemblies`) to specify the path to the directory containing the Assembly References File (`assemblies.xml`) and any third-party assemblies. By default, these files are expected to be in the current working directory of the ABL session. If you specify an invalid directory, the AVM raises an error at startup and halts.

Operating system and syntax	Windows	**`-assemblies`** ***`assemblies-path`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

`assemblies-path`

The absolute or relative path to the directory containing `assemblies.xml` and any third-party assemblies. A relative path is relative to the working directory for the OpenEdge session.

ABL applications using the OpenEdge GUI for .NET must have an `assemblies.xml` file. If the `-assemblies` directory does not contain the file, the AVM raises an error at startup and halts. If you do not specify `-assemblies` and the current working directory does not contain the file, the AVM starts but code that accesses .NET objects generates errors at compile and run time.

If you specify `-assemblies` and the client machine does not have an appropriate version of the .NET framework installed, OpenEdge raises an error and shuts down.

Async Queue Size (-asyncqueuesize)

Use Async Queue Size (`-asyncqueuesize`) to specify the total number of bytes allowed for all buffers allocated when asynchronous requests are queued on behalf of a client. This number determines how many buffers to allocate for each client connection. Each buffer is at most 8KB. If you do not specify `-asyncqueuesize`, the default size for all buffers is 64KB.

Operating system and syntax	UNIX / Windows	**`-asyncqueuesize`** ***`n`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	2,147,483,647	1	—	—

n

The total number of bytes allowed for all asynchronous request buffers allocated for a client.

As each client connects, request buffers are generated and the size of each buffer determined. Both send and receive requests can be queued on a client depending how the program is implemented. When the buffers are filled, additional queued requests are stored in a temporary table associated with that client session. If you set the value of `-asyncqueuesize` to 0, all requests are placed in the temporary table.

Performance may vary based upon whether requests are stored in memory or in the temporary table. So you may want to choose this value based upon how much data you expect to flow between the client and the AppServer for asynchronous requests.

Note: You can use `-asyncqueuesize` only when a client making asynchronous requests is connecting to an AppServer. The client may be an AppServer.

The following example shows how to start a multi-user mode OpenEdge client session using `-asyncqueuesize`:

```
mpro -asyncqueuesize 135000
```

Blocks in Database Buffers (-B)

Use Blocks in Database Buffers (`-B`) to specify the number of blocks in the database buffers. The optimum value depends on your application.

Operating system and syntax	UNIX / Windows	**-B *n***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**[7]
Client Connection, Database Server	1,000,000,000 or 125,000,000[6]	10	20	Maximum value of 3000 or 8 * *users*, whichever is higher

n

The number of blocks in the database buffers.

Blocks in Database Buffers (`-B`) can be changed while the database is online with the PROUTIL INCREASETO utility.

[6] The maximum value of the sum of -B + -B2: 1,000,000,000 for 64-bit platforms; 125,000,000 for 32-bit platforms. The maximum is system dependent and limited by available memory.

[7] The *users* value is specified by the Number of Users (-n) parameter.

Note: On the AIX platform, when starting a database with large shared memory requirements (for instance, when the -B exceeds the allotted system paging space), the system may become unstable if the PSALLOC=early environment variable is not set. For more information, see *OpenEdge Getting Started: Installation and Configuration*.

Note: Release 10.1C and forward, the OpenEdge RDBMS expands the range of internal validations used to ensure database consistency in both index and data blocks during forward processing. Validations using PROUTIL can be run online as part of routine health checks. For more information, see *OpenEdge Data Management: Database Administration*.

Blocks in Alternate Buffer Pool (-B2)

Use Blocks in Alternate Buffer Pool (-B2) to specify the number of blocks in the alternate buffer pool. The optimum value depends on your application.

Operating system and syntax	**UNIX / Windows**	**-B2 *n***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**[9]
CC, DBS	1,000,000,000 or 125,000,000[8]	10	0	0

n

The number of blocks allocated in the alternate buffer pool.

The Alternate Buffer Pool is a collection of buffers in shared memory that are logically separate from the primary buffer pool. Management of the buffers in the Alternate Buffer Pool is independent of the primary buffer pool. Assigning specific database areas or objects to occupy buffers in the Alternate Buffer Pool may improve your buffer hit rate, thereby reducing the need to read and write buffers to and from disk, possibly improving performance.

You cannot specify Blocks in Alternate Buffer Pool for a Replication target database.

Blocks in Alternate Buffer Pool (–B2) can be changed while the database is online with the PROUTIL INCREASETO utility.

For more information on the Alternate Buffer Pool and PROUTIL, see *OpenEdge Data Management: Database Administration*.

[8] The maximum value of the sum of -B + -B2. 1,000,000,000 for 64-bit platforms; 125,000,000 for 32-bit platforms. The maximum is system dependent and limited by available memory.

[9] The *users* value is specified by the Number of Users (-n) parameter.

Batch (-b)

Use Batch (`-b`) to initiate a batch session, with no terminal interaction.

Operating system and syntax	UNIX / Windows	**`-b`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

When ABL (Advanced Business Language) procedures run in batch mode, the ABL Virtual Machine (AVM) does not display any messages produced by those procedures on the terminal screen. However, AVM error messages are displayed to the terminal screen. To suppress these messages, redirect standard output as shown in this Windows example:

```
prowin32.exe -p test.p -b > nul
```

ADE R-code Location (-baseADE)

Use ADE R-code Location (`–baseADE`) to specify where the client should look for another version of ADE r-code on the same machine at the same time.

Operating system and syntax	UNIX / Windows	**`-baseADE`** ***`directory`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

`directory`

Identifies where the client should look for the ADE r-code.

You can have different versions of ADE r-code on the same machine at the same time. As a result, you need a way to specify the ADE r-code directory to the client so that the client can add this directory and all of the procedure libraries contained in the directory to the `PROPATH`. The `–baseADE` startup parameter lets you specify the location of the ADE r-code directory.

At session startup, OpenEdge automatically adds `$DLC/gui` (or `$DLC/tty` for character mode clients) and all procedure libraries in `$DLC/gui` (or `$DLC/tty`) to `PROPATH`.

When you specify `-baseADE`, the `$DLC/gui` (or `$DLC/tty`) directory and the procedure libraries in `$DLC/gui` (or `$DLC/tty`) are not added to `PROPATH`. Instead, the directory specified using the `-baseADE` startup parameter is added to `PROPATH` followed by all of the procedure libraries in the directory. For example, if you use `-baseADE` to specify the following:

```
-baseADE c:\maint\gui
```

And `c:\maint\gui` contains the procedure libraries `adecomm.pl`, `adeuib.pl`, and `protools.pl`, then `PROPATH` looks like the following:

```
...,c:\maint\gui,c:\maint\gui\adecomm.pl,c:\maint\gui\adeuib.pl,c:\maint\gui\protools.pl,...
```

If `$DLC/gui` (or `$DLC/tty`) or any of the procedure libraries from the directory already are part of `PROPATH` (environment variable, `.ini` file, or registry setting), OpenEdge does not remove them.

OpenEdge allows you to specify an empty string for <*directory*>, `-baseADE ""`. In this situation, OpenEdge does not add `$DLC/gui` (or `$DLC/tty`) or the procedure libraries in `$DLC/gui` (or `$DLC/tty`) to `PROPATH`.

You also can specify the ADE R-code Location with the `BASE-ADE` attribute of the `SESSION` system handle. For more information about this `SESSION` attribute, see *OpenEdge Development: ABL Reference*.

Base Index (-baseindex)

Use Base Index (`-baseindex`) with Index Range Size (-indexrangesize) on page 123 to specify the range of indexes for which you want to collect statistics.

Operating system and syntax	**UNIX / Windows**	**`-baseindex` *n***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	—	—	—	—

n

The starting index number in the range of indexes for which you want to track access statistics.

Access to the statistics is handled through the Virtual System Tables (VSTs). Index statistics are stored in the `_IndexStat VST`. To obtain index numbers, use the following ABL code:

```
FOR EACH _file:
  DISPLAY _file._file-name.
  FOR EACH _index WHERE _index._file-recid = RECID(_file):
    DISPLAY _index._idx-num _index._index-name.
  END.
END.
```

Running the preceding code results in the following output:

```
File-Name
filename

  idx-num Index-Name
     n1   index name1
     n2   index name2
     n3   index name3
```

For more information on virtual system tables, see *OpenEdge Data Management: Database Administration*.

Registry Base Key (-basekey)

Use Registry Base Key (`-basekey`) to identify the registry basekey in which to look for environment information.

Operating system and syntax	**Windows**	**`-basekey` *`name`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

`name`

The name of the registry basekey. The following are the valid registry basekey values:

- HKEY_CURRENT_USER
- HKEY_CLASSES_ROOT
- HKEY_LOCAL_MACHINE
- HKEY_USERS
- HKEY_CURRENT_CONFIG
- HKEY_DYN_DATA
- INI

You can force OpenEdge to bypass the registry search by specifying `INI`.

Note: You can use the `-basekey` startup parameter only at the command line. It is ignored when used in a parameter (`.pf`) file.

For more information on using the Registry Base Key (`-basekey)` startup parameter, see *OpenEdge Deployment: Managing ABL Applications*.

Base Table (-basetable)

Use Base Table (`-basetable`) with Table Range Size (-tablerangesize) on page 209 to specify the range of tables for which you want to collect statistics.

Operating system and syntax	UNIX / Windows	**`-basetable`** ***`n`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	—	—	—	—

`n`

The starting table number in the range of tables for which you want to track access statistics.

Access to the statistics is handled through the Virtual System Tables (VSTs). Table statistics are stored in the `_TableStat VST`. To obtain table numbers, use the following ABL code:

```
FOR EACH _file:
  DISPLAY _file._file-num _file._file-name.
END.
```

Running the preceding code results in the following output:

```
_File-Number  File-Name
     n1      table name1
     n2      table name2
     n3      table name3
```

For more information on virtual system tables, see *OpenEdge Data Management: Database Administration*.

Before-image Buffers (-bibufs)

Use Before-image Buffers (`-bibufs`) to specify the number of before-image buffers. This parameter is useful only when running the Before-image Writer (BIW).

Note: Starting in Release 10.1C, the OpenEdge RDBMS expands the range of internal validations used to ensure database consistency in both index and data blocks during forward processing. Validations using PROUTIL can be run online as part of routine health checks. For more information, see *OpenEdge Data Management: Database Administration*.

Operating system and syntax	**UNIX / Windows**	**`-bibufs` *`n`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	—	3	—	20

`n`

The number of before-image buffers.

The BIW continually writes the filled before-image buffers to disk, making the buffers available to other client and server processes. Without a BIW writing the buffers, any gain from increasing the number of buffers is negligible.

Threshold Stall (-bistall)

Use Threshold Stall (`-bistall`) with Recovery Log Threshold (-bithold) on page 79 to quiet the database when the recovery log threshold is reached, without performing an emergency shutdown. When you use `-bistall`, a message is added to the database log (`.lg`) file stating that the threshold stall is enabled.

Operating system and syntax	**UNIX / Windows**	**`-bistall`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	—	—	—	—

Recovery Log Threshold (-bithold)

Use Recovery Log Threshold (`-bithold`) to set the maximum size to which recovery log files can grow.

Operating system and syntax	**UNIX / Windows**	**`-bithold` *`n`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	System dependent[10]	System dependent[10]	—	—

`n`

An `INTEGER` specifying the threshold, in MB.

The recommended threshold is between 3% and 100% of the largest possible recovery log file size, rounded to the nearest cluster boundary. If the threshold is set above 1000MB, OpenEdge issues a warning message to the display and the database log (`.lg`) file. Once the threshold is reached, the database performs an emergency shutdown. See *OpenEdge Data Management: Database Administration* for more information about recovery logs.

Private Read-only Buffers (-Bp)

Use Private Read-only Buffers (`-Bp`) to request a number of private read-only buffers.

Operating system and syntax	**UNIX / Windows**	**`-Bp` *`n`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Connection	64	1	—	—

`n`

An `INTEGER` value greater than 1 but less than 64.

The request remains active for the entire session unless it is changed or disabled by an application. Each user of private read-only buffers reduces the number of public buffers, which is set with Blocks in Database Buffers (-B) on page 72. The total number of private read-only buffers for all simultaneous users is limited to 25% of the total blocks in database buffers.

[10] Limited by available disk space.

Maximum Private Buffers per Connection (-Bpmax)

Use Maximum Private Buffers per Connection (`-Bpmax`) parameter to control the number of private buffers a connection can request through the private read-only buffers (`-Bp`) parameter.

Operating system and syntax	UNIX / Windows	**`-Bpmax`** ***`n`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	25% of -B	1	64	64

`n`

An `INTEGER` specifying the maximum number of private buffers.

The number of blocks in the database buffers is controlled by the blocks in database buffers (`-B`) parameter. Up to 25% of the blocks created may be allocated for use as private buffers. When a client connects to an OpenEdge database, the connection requests `-Bp`*`n`* private buffers. Prior to 9.1C, the maximum value of `-Bp` was 64. With the `-Bpmax` parameter, the broker may override this limit by allowing the value of `-Bp` to be up to 25% of `-B`.

Note: Each use of a private read-only buffer reduces the number of public buffers. The total number of private buffers for all simultaneous users is limited to 25% of the total blocks in database buffers.

Bleeding Record Lock (-brl)

Use Bleeding Record Lock (`-brl`) to let the AVM hold a share lock on a record used by more than one buffer, even after the buffer using the record releases it.

Operating system and syntax	UNIX / Windows	**`-brl`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Connection	—	—	—	—

Browse Column Type (-browcoltype)

Use Browse Column Type (`-browcoltype`) to override the `TYPE` attribute for browse columns, and always return "`BROWSE-COLUMN`" regardless of the actual column type (that is, "`COMBO-BOX`", "`FILL-IN`", or "`TOGGLE-BOX`").

Operating system and syntax	**Windows**	**`-browcoltype`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

For more information on the `BROWSE` widget or the `TYPE` attribute, see *OpenEdge Development: ABL Reference*.

Number of Buffers for Temporary Tables (-Bt)

Use Number of Buffers for Temporary Tables (`-Bt`) to specify the number of buffers in the temporary table database pool.

Operating system and syntax	**UNIX / Windows**	**`-Bt` *`n`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	50,000	10	255	255

`n`

> An `INTEGER` that specifies the number of blocks in the buffer for temporary tables.
>
> OpenEdge uses the value you specify for temporary databases used during the session the same way it uses Blocks in Database Buffers (-B) on page 72 for permanent databases. The default value is 255.

Use Temporary Table Database Block Size (-tmpbsize) on page 212 to specify the temporary table database block size, which dictates the size of each buffer in the temporary table database buffer pool for the current OpenEdge session.

If you do not have enough free memory (virtual or physical) for the buffer pool, reduce the value of `-Bt`. To approximate the amount of memory required (in kilobytes) for the buffer pool, use the following formula:

```
1.1 * (the value of -Bt) * (the value of -tmpbsize)
```

See *OpenEdge Getting Started: ABL Essentials* for more information on temporary tables.

Index Cursors (-c)

Use Index Cursors (`-c`) to specify the number of index place holders, or *cursors*, for an OpenEdge session.

Operating system and syntax	**UNIX / Windows**	**`-c` *`n`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
DataServer	—	—	—	—

`n`

The number of index cursors allowed.

Each active `FOR EACH` and `FIND NEXT` loop uses one index cursor for each index the AVM is using. A cursor takes 64 bytes. If you need more index cursors, the AVM displays the following error message:

```
SYSTEM ERROR: Too few index cursors. Increase -c parameter.
```

Note: The `-c` parameter has a different meaning when you connect to an ORACLE database. For more information, see *OpenEdge Data Management: DataServer for Oracle*.

Schema Cache File (-cache)

Use Schema Cache File (`-cache`) to read the database schema from a local file instead of the database. You must have previously built the schema cache and stored it as a binary file.

Operating system and syntax	**UNIX / Windows**	**`-cache` *`filename`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Connection	—	—	—	—

`filename`

The pathname of a binary schema cache file.

To perform database activities, the OpenEdge client keeps a copy of the database schema called the *schema cache* in memory. By default, OpenEdge creates the schema cache by reading the database schema stored in the database file. The time required to read the schema usually is minimal; however, under the following conditions, the time required to read the schema might be unacceptable:

- If the client connects to the database over a wide-area network (WAN)
- When a large number of clients connect to a database simultaneously, for example, after a database shutdown or crash

Connection time depends on several factors, including schema size.

To reduce connection time, OpenEdge lets you store the schema cache as a binary file, called a *schema cache file*, on a local disk. The client can then read the schema directly from the schema cache file.

To write the schema cache file, you build the desired schema cache and save it to a binary file using the ABL `SAVE CACHE` statement. The schema cache file is portable across systems, so you can create the file once and distribute it across a heterogeneous network of systems. For information on building and saving the schema cache file, see *OpenEdge Development: Programming Interfaces*.

If you specify schema cache file (`-cache`) when you connect to a database and the local schema is valid, the AVM reads the schema from the local file instead of from the database. The schema cache is valid if the time stamp of the schema cache file matches the time stamp in the database master block. If the time stamps do not match, or for some reason the AVM cannot read the file, the AVM issues a warning message and reads the schema from the database.

Note: If you are generating the local binary schema cache, do not connect to the database using Trigger Location (-trig) on page 213 and Schema Cache File (`-cache`) together.

Check Double-byte Enabled (-checkdbe)

Use Check Double-byte Enabled (`-checkdbe`) when you modify an OpenEdge application to support double-byte characters or extended character data.

Operating system and syntax	**UNIX / Windows**	**`-checkdbe`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

The `-checkdbe` parameter specifies that ABL generate compile-time warning messages whenever it finds a `LENGTH` function, `SUBSTRING` function, `SUBSTRING` statement, or `OVERLAY` statement without a `CHARACTER`, `COLUMN`, `RAW`, or `FIXED` option.

Check Width (-checkwidth)

Use Check Width (`-checkwidth`) to specify whether the AVM compares `CHARACTER`, `DECIMAL`, and `RAW` field data against the metaschema `_width` field value before updating a database record. The `_width` field value specifies the maximum width of the data allowed in a field.

Operating system and syntax	**UNIX / Windows**	**`-checkwidth`** ***n***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

n

The check-width mode. Valid check-width modes are:

- **0** — Ignore the `_width` field value and store the data. This is the default mode.
- **1** — Store the data, and generate a warning message if the data exceeds the size specified in the `_width` field.
- **2** — Do not store data that exceeds the size specified in the `_width` field, and generate an error. Specify this mode if you want ABL to behave like SQL.

ABL allows for variable length data; SQL does not. This startup parameter lets you impose the SQL requirement that data cannot exceed the size specified in the `_width` field.

Note: The `_width` field value for array fields is the size of all array elements after the AVM converts the array element values to `CHARACTER`. This means that even with the `-checkwidth` startup parameter enabled, array fields can still surpass the `_width` field value.

SQL Server Java Classpath (-classpath)

Use SQL Server Java Classpath (`-classpath`) to identify the Java classpath to use when starting an SQL server.

Operating system and syntax	**UNIX / Windows**	**`-classpath` *`pathname`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
SQL	—	—	—	—

`pathname`

Specifies the pathname of the classpath.

SQL database brokers use this parameter when launching the Java Virtual Machine (JVM) to execute stored procedures. The default is to use the current environment variable `CLASSPATH` setting. You do not use this parameter directly.

Clear Log (-clearlog)

Use Clear Log (`-clearlog`) to delete any file that matches the name of the specified client log file identified by the Client Logging (-clientlog) on page 86 parameter.

Operating system and syntax	**UNIX / Windows**	**`-clearlog`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

Note: This parameter is valid only for interactive or batch clients. WebSpeed agents and AppServers will silently ignore it.

In the following example, the client will delete the log file `debug.log`, if it exists, and then open `debug.log` as the client log file:

```
prowin32.exe -p myprog.p -clientlog debug.log -clearlog
```

In the following example, the client will delete all the log files matching the name `debug.log`; such as `debug.000001.log` through `debug.000003.log`. Then the client will open `debug.000001.log` as the client log file, as shown:

```
prowin32.exe -p myprog.p -clientlog debug.log -clearlog -logthreshold 1000000
```

WebSpeed agents and AppServers will silently ignore the Client Logging (-clientlog) on page 86 parameter. The broker handles the clearing of the WebSpeed and AppServer server logs, through the `srvrLogAppend` property in the `ubroker.properties` file.

For more detailed information about enabling logging, see *OpenEdge Development: Debugging and Troubleshooting*.

Client Logging (-clientlog)

Use Client Logging (`-clientlog`) to allow an application to automatically write error and warning messages to the specified log file.

Operating system and syntax	**UNIX / Windows**	**`-clientlog` *`filename`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

`filename`

The pathname and filename of the log file OpenEdge uses for messages, ABL stack trace, and .NET stack trace information.

If the filename you supply is a relative pathname, then a file is accessed relative to the current working directory. If the filename is an absolute pathname, then the specified file is accessed.

Note: Do not include a numbered sequence in the filename. This might conflict with the rolled over log files OpenEdge creates based on your Number of Log Files to Keep (-numlogfiles) on page 169 and Log Threshold (-logthreshold) on page 140 startup parameter settings.

Use the Log Entry Types (-logentrytypes) on page 134 startup parameter to specify one or more types of log entries you want to write to the log file. Use the Logging Level (-logginglevel) on page 139 startup parameter to specify the level at which log entries are written to the log file.

When you use the -clientlog startup parameter, and you also specify the Debug Alert (-debugalert) on page 104 startup parameter or set SESSION:DEBUG-ALERT to TRUE, the log file includes an ABL stack trace for error messages (ABL errors and .NET Exceptions) and Alert-box messages. The top of the stack (most recent call) is displayed at the top of the trace listing.

.NET Exceptions can be thrown when working with .NET objects in ABL. When you do not handle these Exceptions—there is no CATCH or NO-ERROR logic present—and Debug Alert is on, the AVM adds the .NET stack trace to the Debug Alert information after the ABL call stack. The ABL stack trace and the .NET stack trace are added both in the Debug Alert Help dialog box and in the client log (when -clientlog is specified).

If an error message is diverted to the ERROR-STATUS system handle, and client logging is enabled, then no information is written to the log file.

In a non-interactive session, the application is configured so that the output device is associated with a file (or another device). In this configuration, when an ABL statement encounters an error, it writes the error to the output device. If client logging is enabled, then this message is also written to the specified log file. Errors are written to the output device at logging level 1 (Error) and up.

You can use the MESSAGE statement with the VIEW-AS ALERT-BOX option to write application specific information to the screen and the log file. In this case, you must specify an entry type of "4GLMessages" and a logging level of 2 (Basic), at least.

Note: When you specify a non-zero value for the Log Threshold (-logthreshold) on page 140 startup parameter, only one client process at a time can open the log file. Therefore, consider specifying a different log file for each client session.

For more information about logging levels, see the Log Entry Types (-logentrytypes) on page 134 startup parameter reference entry. For more information about specifying log entry types, see the Log Entry Types (-logentrytypes) on page 134 startup parameter reference entry. For more information about -dslog startup parameter, refer to the DataServer Logging (-dslog) on page 108.

You can use attributes on the LOG-MANAGER system handle to specify log entry types and logging levels in the client context. You can also use attributes on the DSLOG-MANAGER system handle to specify log entry types and logging levels in the server context.

For more information about the ABL elements previously referenced above, see *OpenEdge Development: ABL Reference*.

For more detailed information about enabling logging, see *OpenEdge Development: Debugging and Troubleshooting*.

Cluster Mode (-cluster)

Use Cluster Mode (-cluster) to startup a cluster-enabled database.

Operating system and syntax	UNIX / Windows	**-cluster *qualifier***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	—	—	—	—

qualifier

Direct a broker to start a cluster-enabled database directly, or to instruct the cluster manager to perform the startup.

Note: All cluster-enabled databases require this parameter. If used with a database that is not cluster-enabled, the parameter is ignored.

The following table describes the cluster mode qualifier.

Table 1: Cluster mode qualifier values

Value	Description
`startup`	Instructs the broker to redirect the startup request to the OS cluster manager. The cluster manager will start the database using the information supplied when the database was enabled for cluster protection.
`protected`	Instructs the broker to start a cluster-enabled database. This qualifier value for `-cluster` is specified within the database `.pf` file used by the cluster manager, and indicates that the database will be protected by the cluster manager in the event of a fail-over. **Note:** This qualifier value must only be used by the cluster manager. It should not be used on the command line or a directly executed script.

For more information on the cluster manager, see *OpenEdge Data Management: Database Administration*.

Combo-box Font (-combofont)

Use the Combo-box Font (`-combofont`) startup parameter to change the default font of decimal values displayed in combo-boxes with `LIST-ITEM-PAIRS`.

Operating system and syntax	**UNIX / Windows**	**`-combofont [ FIXED ]`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

`FIXED`

Specifies that the default font for decimal combo-boxes with LIST-ITEM-PAIRS is fixed.

In Release 9.1D, a combo-box representing a decimal field or variable with LIST-ITEM-PAIRS displays values using a fixed font, by default. Starting with Release 9.1D07, you can use the `-combofont` startup parameter (without the `FIXED` option) to change the default font to a proportional font.

Starting with Release 10.0A, a decimal combo-box with LIST-ITEM-PAIRS displays values using a proportional font, by default. To change the default font back to a fixed font, specify the `-combofont` startup parameter with the `FIXED` option.

A decimal combo-box with LIST-ITEMS displays values using a fixed font, by default. If you change a decimal combo-box with LIST-ITEMS to LIST-ITEM-PAIRS, after it is realized, the font automatically changes from fixed to proportional (unless you specified the `-combofont` startup parameter with the `FIXED` option).

Note: If you specify a font in the FONT attribute for a combo-box, specifying this startup parameter has no effect on the combo-box font.

Conversion Map (-convmap)

Use Conversion Map (`-convmap`) to identify the `CONVMAP` file to use for code page conversions, collation orders, and case conversions.

Operating system and syntax	**UNIX / Windows**	**`-convmap` *`filename`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session, Database Server	—	—	—	—

`filename`

Pathname of your `CONVMAP` file.

By default, OpenEdge uses the `convmap.cp` file in the `DLC` directory. You can create a `CONVMAP` file by using the PROUTIL utility with the CODEPAGE-COMPILER qualifier. See *OpenEdge Development: Internationalizing Applications* for more information on `CONVMAP` files.

Communications Parameter File (-cp)

Use Communications Parameter File (`-cp`) to override preset connection parameters for auto-connect databases.

Operating system and syntax	UNIX / Windows	**-cp { *com-parms* | *filename* }**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Connection	—	—	—	—

com-parms, filename

The database connection parameters or parameter file.

Communications parameters supplied with the `-cp` parameter override any connection information stored in the OpenEdge metaschema field `_Db-comm` in the `_Db` file record for the database.

Case Table (-cpcase)

Use Case Table (`-cpcase`) to specify the case table.

Operating system and syntax	UNIX / Windows	**-cpcase *tablename***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session, Database Server	—	—	Basic	Basic

tablename

Name of a case table in the `convmap.cp` file.

This table establishes case rules for the code page that the AVM uses in memory. The code page is specified by the Internal Code Page (-cpinternal) on page 91 parameter. The case rules are used by the CAPS and LC functions. Also, in a character field format, you can use an exclamation point (`!`) to tell the AVM to convert all characters to uppercase during input.

To retrieve the value of this startup parameter at runtime, use the `SESSION` system handle.

Collation Table (-cpcoll)

Use Collation Table (`-cpcoll`) to identify a collation for the AVM to use with the code page in memory. The code page is specified by the Internal Code Page (-cpinternal) on page 91 parameter.

Operating system and syntax	**UNIX / Windows**	**`-cpcoll`** ***`collation-name`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session, Database Server	—	—	Basic	Basic

`collation-name`

The name of a collation table within the `convmap.cp` file, or the name of a collation in the International Components for Unicode (ICU) library.

The AVM uses the collation rules that you specify to compare characters and sort records if a BY clause cannot be satisfied by an index. The collation rules specified with the `-cpcoll` parameter take precedence over the collation specified for any database the AVM accesses during the session, except when the AVM uses or modifies pre-existing indexes. When you update or rebuild a database's indexes, the AVM uses the collation rules originally defined for that database.

If you do not use `-cpcoll`, the AVM uses the language collation rules defined for the first database on the command line. If you do not specify a database on the command line, the AVM uses the collation rules with the name "basic."

To retrieve the value of this startup parameter at runtime, use the `SESSION` system handle.

See *OpenEdge Development: Internationalizing Applications* for more information on collations.

Internal Code Page (-cpinternal)

Use Internal Code Page (`-cpinternal`) to identify the code page that the AVM uses in memory and for graphical clients.

Operating system and syntax	**UNIX / Windows**	**`-cpinternal`** ***`code-page`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session, Database Server, DataServer	—	—	iso8859-1	iso8859-1

code-page

Name of the code page that the AVM uses in memory.

For graphical clients, the `-cpinternal` code page should be the same code page that the operating system uses. If you do not use `-cpinternal`, the AVM defaults to the iso8859-1 code page.

Note: Do not use a 7-bit table with `-cpinternal`. Only use 7-bit tables for converting data from a 7-bit terminal to another code page. Do not use them for character conversion in memory, or for the database.

To retrieve the value of this startup parameter at runtime, use the `CPINTERNAL` attribute of the `SESSION` system handle.

Log File Code Page (-cplog)

Use Log File Code Page (`-cplog`) to identify the code page that OpenEdge uses to write messages to the log (`.lg`) file. If you do not specify a value, the default is the code page specified by Stream Code Page (-cpstream) on page 95.

Operating system and syntax	**UNIX / Windows**	**`-cplog` *`code-page`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session, Database Server, DataServer	—	—	-cpstream	-cpstream

code-page

Name of the code page for messages written to the log file.

To retrieve value of this startup parameter at runtime, use the `CPLOG` attribute of the `SESSION` system handle.

Print Code Page (-cpprint)

Use Print Code Page (`-cpprint`) to identify the code page OpenEdge uses when it prints. When you print a file, the code page specified by `-cpprint` overrides the code page specified by Stream Code Page (-cpstream) on page 95.

Operating system and syntax	UNIX / Windows	**`-cpprint`** ***`code-page`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session, Database Server, DataServer	—	—	-cpstream	-cpstream

`code-page`

Name of the code page used for printer output.

To retrieve the value of this startup parameter at runtime, use the `SESSION` system handle.

R-code in Code Page (-cprcodein)

Use R-code in Code Page (`-cprcodein`) to read the r-code text segments, as if they were written in the code page specified by `-cprcodein`, and convert them to the Internal Code Page (-cpinternal) on page 91 code page. Usually when OpenEdge reads r-code, it converts text segments to the code page specified by Internal Code Page (-cpinternal) on page 91.

Operating system and syntax	UNIX / Windows	**`-cprcodein`** ***`code-page`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session, Database Server, DataServer	—	—	-cpinternal	-cpinternal

`code-page`

Name of the code page for reading r-code text segments.

Caution: This parameter is for use during very rare situations and in general should not be used. OpenEdge reads text segments as if they are written in the code page specified by R-code In Code Page (-cprcodein), even if the text segments were written with a different code page. For example, if you use the following startup parameters and run a `.r` file written with code page IBM850, OpenEdge converts the text segments from ISO8859-1 to ibm861. This can produce incorrect results, if the `.r` file was correctly labeled internally as IBM850: `-cprcodein ISO8859-1 -cpinternal ibm861`

To retrieve the value of this startup parameter at runtime, use the `SESSION` system handle. To determine the code page of an r-code file, use the `RCODE-INFO` handle.

R-code Out Code Page (-cprcodeout)

Use R-code Out Code Page (`-cprcodeout`) to direct OpenEdge to use the code page you specify to mark and convert text segments (any translatable text) when it writes r-code. If you do not specify `-cprcodeout`, OpenEdge uses the code page you specify with Internal Code Page (-cpinternal) on page 91 to write r-code.

Operating system and syntax	**UNIX / Windows**	**`-cprcodeout` _`code-page`_**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	-cpinternal	-cpinternal

`code-page`

Name of the code page for writing r-code text segments.

Typically, you do not need to specify this parameter since OpenEdge converts the text segment to the `-cpinternal` code page when reading r-code. However, if the code page of the intended user is known, you can use this parameter to provide r-code in the user's code page. Performance savings are not significant.

To retrieve the value of this startup parameter at runtime, use the `SESSION` system handle. To determine the code page of an r-code file, use the `RCODE-INFO` handle.

Stream Code Page (-cpstream)

Use Stream Code Page (`-cpstream`) to identify the code page OpenEdge uses for stream I/O. Character terminals use the code page you specify for `-cpstream` unless you also specify a value for Terminal Code Page (-cpterm) on page 96, Print Code Page (-cpprint) on page 93, or Log File Code Page (-cplog) on page 92.

Operating system and syntax	**UNIX / Windows**	**`-cpstream` *`code-page`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session, Database Server, DataServer	—	—	ibm850	ibm850

`code-page`

Name of the code page for stream I/O.

Stream I/O consists of the following elements:

- Terminals (includes character terminals and DOS Protected mode, but does not include graphical interfaces or the Windows character interface)
- Data (`.d`) files
- `READ-FILE`, `WRITE-FILE`, and `INSERT-FILE` methods for the `EDITOR` widget
- `INPUT FROM` and `OUTPUT TO` statements
- All compilable files, such as: `.p`, `.w`, `.i`
- Compiler-generated `LISTING`, `XREF`, and `PREPROCESS` files

Note: Do not use a 7-bit table with -cpstream. Use 7-bit tables for converting data from a 7-bit terminal to another code page only. Do not use them for character conversion in memory or for the database.

To retrieve the value of this startup parameter at runtime, use the `SESSION` system handle. To determine the code page of an r-code file, use the `RCODE-INFO` handle.

Terminal Code Page (-cpterm)

Use Terminal Code Page (`-cpterm`) to identify the code page of your character terminals. This parameter allows you to specify a different code page for character terminals than used by the rest of stream I/O, which is set by Stream Code Page (-cpstream) on page 95.

Operating system and syntax	UNIX / Windows	**-cpterm** ***code-page***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session, Database Server, DataServer	—	—	-cpstream	-cpstream

code-page

Name of the code page for character terminals.

Note: You can use a 7-bit table with -cpterm.

To retrieve the value of this startup parameter at runtime, use the `SESSION` system handle. To determine the code page of an r-code file, use the `RCODE-INFO` handle.

Crash Recovery Status (-crStatus)

Use Crash Recovery Status(`-crStatus`) to specify how often a crash recovery progress message is generated during crash recovery.

Operating system and syntax	UNIX / Windows	**-crStatus** ***n***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Connection, Client Session, Database Server	1800	0	30	30

n

The time, in seconds, between crash recovery progress messages.

Specify 0 to turn off crash recovery progress messages.

Each crash recovery progress message contains the following details:

- The current crash recovery phase
- An estimate of the progress of the current crash recovery phase, expressed as the percent complete
- The current block being processed
- The total number of elapsed seconds for the current phase
- Optionally, if `-crTXDisplay` is also specified, the counter of the BI cluster. For more information, see Crash Recovery Transaction Display (-crTXDisplay) on page 97

For more information about crash recovery, see *OpenEdge Data Management: Database Administration*.

Crash Recovery Transaction Display (-crTXDisplay)

Use Crash Recovery Transaction Display (`-crTXDisplay`) to display to the screen only (not the log file) the contents of the transaction table prior to the Physical Undo phase of crash recovery.

Operating system and syntax	**UNIX / Windows**	**-crTXDisplay**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Connection, Client Session, Database Server	–	–	–	–

When `-crTXDisplay` is specified, the progress messages of crash recovery, controlled by `-crStatus`, includes the counter of the current BI cluster. Use this information to cross-reference the transaction table to determine the location in the BI file of the current phase, relative to the remaining transactions. This information can assist you in determining if there are many transactions to back out during crash recovery, or if there is perhaps one long-running transaction.

The transaction display message contains the following:

- The transaction ID
- The timestamp when the transaction began
- The user number that initiated the transaction
- The counter of the BI cluster that was current when the transaction began

For more information about crash recovery progress messages, see Crash Recovery Status (-crStatus) on page 96.

For more information about crash recovery, see *OpenEdge Data Management: Database Administration*.

Cursor Size (-cs)

Use Cursor Size (`-cs`) to change the maximum number of index levels. You seldom need to use this parameter.

Operating system and syntax	**UNIX / Windows**	**`-cs` *n***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Connection	256	1	6	6

n

The cursor size, which is the number of index levels.

OpenEdge indexes have a tree structure. Even large indexes typically are no deeper than three or four levels. Index cursors, which act as place holders during file access, have size 6 by default; that is, they support up to five levels of indirect access. Therefore, few users ever require the cursor size (`-cs`) parameter. However, in rare cases involving large index keys, increase `-cs` in response to the following error message:

```
Index index-number has at least n levels. Increase -cs parameter.
```

Increasing `-cs` by 2 should be sufficient.

Connection Retry Attempts (-ct)

Use Connection Retry Attempts (`-ct`) to define the number of retries for a database connection attempt.

Operating system and syntax	**UNIX / Windows**	**`-ct` *n***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Connection	-	1	50	-

n

The number of database connection retry attempts. Specify a value greater than 0.

Compile Warning List (-cwl)

Use Compile Warning List (`-cwl`) to alert users to the occurrence of specified statements in the source code. For example, you might want to flag the occurrence of deprecated statements, to discourage their use in favor of new alternative statements.

Operating system and syntax	**UNIX / Windows**	***-cwl filename***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

`filename`

The name of a file that contains ABL statements that are to trigger a warning if they are found in the source code at compile time. Each statement must use a single line in the file. OpenEdge uses the `PROPATH` variable to locate *`filename`*. If it cannot find *`filename`*, it displays an error message and does not start.

When source code containing one or more instances of the specified statements is compiled, OpenEdge displays a message box containing a warning for each such instance, in the following format:

```
The  flagged_statement  statement from the compile warning list
  was found in file  file_name  at line  line_number .
```

Where:

- *`flagged_statement`* is the statement matching an entry in the Compile Warning List file
- *`file_name`* is the source file containing the flagged statement
- *`line_number`* is the line in the source file where the flagged statement occurs

The user can click **OK** in the message box to continue. Compilation occurs normally

You can specify that a warning should occur only if the statement includes specific phrases or options. To do so, add those phrases or options to the entry in the list. For example, the following entry results in a warning only for instances of the `DEFINE FRAME` statement that include the `SHARED` option:

```
DEFINE SHARED FRAME
```

In the preceding example, simple `DEFINE FRAME` statements do not raise a warning message. However, the statement `DEFINE NEW SHARED FRAME` does result in a warning; even though the syntax does not exactly match the list entry, OpenEdge recognizes it as an instance of the flagged `DEFINE SHARED FRAME` statement.

Statements found in comments do not trigger warning messages.

Directory Size (-D)

Use Directory Size (-D) to change the number of compiled procedure directory entries.

Operating system and syntax	**UNIX / Windows**	**-D *n***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	2,147,483,647	5	100	100

n

The number of compiled procedure directory entries.

Each compiled procedure executed during an OpenEdge session requires a directory entry. A compiled procedure can be a session-compiled version of a procedure or a precompiled r-code version.

When the AVM creates a session-compiled version of a procedure and there is no available space in the directory, it discards the oldest inactive compilation of a procedure. The next time the discarded procedure is run, the AVM must recompile it if it was a session compile, or reopen and reread it if it was a precompiled r-code version.

The -D limit is a soft limit. If your application needs to exceed the limit, OpenEdge automatically increases the number of directory entries by 50 percent and dynamically allocates memory for the expanded array. (Use the Statistics (-y) on page 224 parameter to check the current directory size (-D) value.) When the limit is reached, OpenEdge issues a WARNING message, which is written to the current output (if there is current output) and to the log file (LG).

You can force OpenEdge to adhere to the specified directory size (-D) limit by starting the session with the startup parameter. When you use the -hardlimit startup parameter, OpenEdge issues a WARNING message when you exceed the directory size limit. It also issues a message that a resource limit was reached and raises an untrappable STOP condition. Note that -hardlimit also enforces the limits set by the Local Buffer Size (-l) on page 129, the Maximum Memory (-mmax) on page 150, and the Nested Blocks (-nb) on page 157 startup parameters, so the STOP condition is raised when any of the specified limits are reached.

Date Format (-d)

Use Date Format (`-d`) to specify the format used to display dates in an application using a three-character string of the letters: `d`, `m`, and `y` in any order. This startup parameter provides the same functionality as the SESSION:DATE-FORMAT attribute.

Operating system and syntax	**UNIX / Windows**	**`-d` *`dateform`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	mdy	mdy

`dateform`

The format used to display dates in an application.

Note:

- The `-d` sets the `display` format not the `storage` format. The storage format is fixed.
- The date constants entered in procedures, or as initial values in the Data Dictionary, are always specified in month/day/year format.

DataService (-DataService)

Use DataService (`-DataService`) to connect through a NameServer to an ODBC, ORACLE, or SQL Server DataServer.

Operating system and syntax	**UNIX / Windows**	**`-DataService` *`data-service`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Connection	—	—	—	—

`data-service`

The name of the DataService.

The value for *`data-service`* must be a valid name from the DataService list registered to this NameServer, as defined by your appServiceName List entry in the broker properties file. If a default DataService has been defined for your broker instance, you can omit this parameter and connect using the default service.

This parameter must be used in conjunction with the DataServer Logging (-dslog) on page 108 startup parameter.

Note: Using the `-DirectConnect` parameter on the `CONNECT` statement will override the `-DataService` parameter.

For more information on the `CONNECT` statement, see *OpenEdge Development: ABL Reference*.

Physical Database Name (-db)

Use Physical Database Name (`-db`) to connect one or more OpenEdge databases.

Operating system and syntax	**UNIX / Windows**	**`-db` *`physical-dbname`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Connection	—	—	—	—

`physical-dbname`

The database to connect to when an OpenEdge session is started. Specify the **physical** (not the **logical**) database name. If you are connecting in single-user mode or through shared memory, and the database is not in your current directory, you must specify the full path of the database.

Use Single-user Mode (-1) on page 65 after the name of the database to connect in single-user mode. You do not have to precede the first database on the command line with `-db`; however, for all subsequent databases that you want to connect, precede the database name with `-db`. For example, the following command connects two databases, `abc` and `xyz`:

```
pro abc -B 30 -db xyz -B 40
```

Database consistency check (-DbCheck)

Use Database consistency check to enable consistency checking for all record and index blocks in the entire database.

Operating system and syntax	**UNIX / Windows**	**`-DbCheck`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Connection, Database Server	—	—	—	—

When enabled, this option applies consistency checks to all index blocks and record blocks for record and index write operations. Database consistency check validates a block is still physically correct after an operation has been performed. For example, after an index key is inserted into an index block, the consistency check validates that the block still laid out correctly.

You can limit the scope of consistency checking by specifying one area with Area consistency check (-AreaCheck) on page 70, one table with Table consistency check (-TableCheck) on page 208, or one index with Index consistency check (-IndexCheck) on page 122.

Database consistency checking can be enabled or disabled while your database is online with PROMON. See *OpenEdge Data Management: Database Administration* for more information.

Database consistency check can be enabled for a single user client or for any offline utility. If the utility is read-only, such as DBANALYS, the `-DbCheck` parameter is ignored. Online utilities determine whether or not to perform database consistency checking during execution based on the use of `-DbCheck` at broker startup or by the enablement/disablement of the database consistency checking in PROMON.

Debugger (-debug)

Use Debugger (`-debug`) to start an OpenEdge session by running the Application Debugger in stand-alone mode.

Operating system and syntax	**UNIX / Windows**	**`-debug`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

This makes the Debugger (not the Procedure Editor) the point of entry for all application procedures to run. The Debugger begins by allowing breakpoints before using the Debugger `RUN` command to execute the startup procedure. Also, the AVM automatically sets a breakpoint on the first executable line of the startup procedure before it runs.

For more information on the Debugger, see *OpenEdge Development: Debugging and Troubleshooting*.

Debug Alert (-debugalert)

Use Debug Alert (`-debugalert`) to access ABL and .NET stack trace information during a session. Its major use is for error alert boxes, so you can figure out where an error has occurred in your code.

Operating system and syntax	**UNIX / Windows**	**`-debugalert`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

Alternately, you can set the `DEBUG-ALERT` attribute on the `SESSION` system handle to `TRUE` in your application. The advantage of the `DEBUG-ALERT` attribute is that its value can be modified during a client session.

When an unhandled ABL error or .NET Exception occurs—there is no `CATCH` or `NO-ERROR` logic present—in an interactive session for any ABL client, the AVM displays an Error message box. When Debug Alert is `TRUE`, the message box also includes a `Help` button. Clicking on the `Help` button produces a `Stack Trace` dialog box containing ABL stack trace and .NET stack trace information. If the error was generated by throwing an error object from the ABL code and the Error Stack (`-errorstack`) startup parameter is used, or the `ERROR-STACK-TRACE` attribute on the `SESSION` system handle is set to true, then the stack trace also includes the contents of the `CallStack` property for the unhandled error. The stack trace information is written to the client log when the Client Logging (`-clientlog`) startup parameter is specified. The top of the stack (most recent call) is displayed at the top of the trace listing.

To retrieve or reset the value of this parameter at runtime, you use the `DEBUG-ALERT` attribute of the `SESSION` system handle. For more information on the `DEBUG-ALERT` attribute or the `SESSION` system handle, see *OpenEdge Development: ABL Reference*.

Enable Attachable Debugging (-debugReady)

Use Enable Attachable Debugging (`-debugReady`) to enable an OpenEdge process (that is, an ABL client, a single WebSpeed agent, or single AppServer process) to be attached to by the Debugger.

Operating system and syntax	**UNIX / Windows**	**`-debugReady`** { **`0`** | ***`port-number`*** }		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

`port-number`

> The port on which the AVM opens a socket for connecting to the attachable Debugger. If you want the AVM to find an available port, enter 0 (zero). The default value is 0.

An ABL developer can attach the attachable Debugger to any OpenEdge session running locally or remotely in a LAN environment. This allows the developer to run the Debugger on one machine and debug an OpenEdge process running on another machine.

You cannot specify both the Debugger (-debug) on page 103 and Enable Attachable Debugging (`-debugReady`) startup parameters on the command line at the same time. If you do, the AVM generates an error and shuts down.

For more information about the attachable Debugger, see *OpenEdge Development: Debugging and Troubleshooting*.

Default scrolling (-defaultscrolling)

Apply the SCROLLING option to all DEFINE QUERY STATIC statements in order to create client-side result lists. This option improves performance when query results are received over a network. However, performance is reduced if there is no networking.

Operating system and syntax	**UNIX / Windows**	**`-defaultscrolling`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

When the SCROLLING option is enabled for DEFINE QUERY STATIC statements, the server can pack multiple records in a single network message. This is known as *prefetching*. Prefetching allows for repositioning to other records in the result set on the client side rather than on the server side. Therefore, prefetching can significantly reduce network traffic.

Note:

- Source code is not recompiled when you use the `-defaultscrolling` startup parameter.
- If there is a sort or preselect operation, the `-defaultscrolling` startup parameter only affects the presort pass, when the rowids and sort fields are collected.
- The `-noautoreslist` startup parameter conflicts with `-defaultscrolling`. If both parameters are specified, a runtime error results.

Disable Delete Trigger (-disabledeltrig)

Use Disable Delete Trigger (`-disabledeltrig`) to disable the delete trigger when ALLOW-REPLICATION is on for the DISABLE TRIGGERS statement or the DISABLE-LOAD-TRIGGER() method of a buffer. In this case the replication triggers run, but the delete trigger does not.

Operating system and syntax	**UNIX / Windows**	**`-disabledeltrig`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

Dictionary Expressions (-dictexps)

Use Dictionary Expressions (`-dictexps`) to specify that ABL use dictionary and help validation for all fields in all frames when you compile an application. ABL continues compiling even after it encounters a validation error.

Operating system and syntax	**UNIX / Windows**	**`-dictexps`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

This parameter serves as a temporary step to close possible validation holes in existing applications. It has the effect of adding a `USE-DICT-EXPS` option to every frame in the application, including those that are not used for input. This parameter closes all possible validation holes, but it is highly inefficient.

You can override this functionality for a particular frame by using the `NO-VALIDATE` keyword for that frame. See the Frame Phrase reference entry in *OpenEdge Development: ABL Reference* for more details on the `NO-VALIDATE` option.

If you do not specify `-dictexps`, ABL uses dictionary and help validation for a field only if that field is used in an `UPDATE`, `SET`, `PROMPT-FOR`, or `ENABLE` statement. In addition, ABL does not use dictionary and help validation when compiling if you specify only the *`widget`*`:SENSITIVE=YES` construct to make a widget sensitive. ABL stops compiling when it encounters a validation error.

Direct I/O (-directio)

Use Direct I/O (`-directio`) to open all files in unbuffered mode, which enables OpenEdge to use an I/O technique that bypasses the operating system buffer pool and transfers data directly from a buffer to disk.

Operating system and syntax	**UNIX / Windows**	**`-directio`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Connection, Database Server	—	—	Not enabled	Not enabled

This technique has several advantages over buffered reads and writes, such as avoiding the overhead of maintaining the operating system buffer pool and eliminating competition for operating system buffers between OpenEdge programs and other programs. The operating system buffer-pool algorithms are designed for efficient sequential file access; the OpenEdge buffer-pool algorithms are more efficient for access to an OpenEdge database.

You might improve OpenEdge performance by using the direct I/O feature. To use direct I/O, use Blocks in Database Buffers (-B) on page 72 to increase the size of the OpenEdge buffer pool, since OpenEdge I/O does not pass through the operating system buffer pool. Also, decrease the size of the operating system buffer pool to compensate for the additional memory allocated to OpenEdge.

Note: Use asynchronous page writers (APWs). They improve database performance by performing overhead operations in the background.

DataServer Logging (-dslog)

Use DataServer Logging (`-dslog`) to allow an application to automatically write server error and warning messages, and other server log messages, to the specified log file name. You can apply the `-dslog` startup parameter to the DataServer for MS SQL Server as well as the DataServer for Oracle.

Operating system and syntax	**UNIX / Windows**	**`-dslog` *`filename`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
DataServer Session	—	—	—	—

`filename`

The name of log file the DataServer uses for log messages.

The `—dslog` startup parameter specifies the DataServer log file name.

If you do not specify a log file name at startup, then the default `dataserv.lg` file will be used for logging server information.

The `dataserv.lg` file is the default value for the `-dslog` parameter. For non-interactive sessions that encounter errors, warnings or other messages that need to be logged, it writes to this log file by default.

If the filename you supply is a relative pathname, then a file is accessed relative to the current working directory. If the filename is an absolute pathname, then the specified file is accessed unless the `$DSLOGDIR` environment variable is set to specify a directory path for the `-dslog` file name. The `$DSLOGDIR` setting overrides the current working directory in search for the `-dslog` file name.

Note: Do not include a numbered sequence in the filename. This might conflict with the rolled over log files OpenEdge creates based on your Number of Log Files to Keep (-numlogfiles) on page 169 and Log Threshold (-logthreshold) on page 140 startup parameter settings.

Use the Log Entry Types (`-Dsrv logentrytypes`) DataServer connection parameter to specify one or more types of log entries you want to write to the server context log file. Use the Logging Level (`-Dsrv logginglevel`) DataServer connection parameter to specify the level at which log entries are written to the log file.

ABL clients, including the client component of the DataServer, write their logging to the file specified by the `—clientlog` startup parameter. For more information about the `-clientlog` startup parameter, see Client Logging (-clientlog) on page 86.

For more information about Log Entry types in the client and server contexts, see Log Entry Types (-logentrytypes) on page 134.

The OpenEdge client, including the client component of the DataServer, to which the `-clientlog` switch applies, can be thought of as the "client log context." Similarly, the server component of the DataServer, to which the `-dslog` switch applies, can be considered the "server log context." When the OpenEdge DataServer runs self-service (i.e., not client/server), the OpenEdge client's log context and the OpenEdge DataServer's log context can be merged into a single log file by selecting the same file name with the `-clientlog` and `-dslog` log file specifications in both log contexts. Note that the `-logthreshold`, `-logappend` and `-numlog` file parameters specified at client startup apply to both client and server log contexts in this case.

Note: If the`-numlogfiles` startup parameter is used without a specification for the `-logthreshold`, then the `-numlogfiles` startup parameter is ignored. See Log Threshold (-logthreshold) on page 140 and Number of Log Files to Keep (-numlogfiles) on page 169 for more details.

The `-dslog`, `-logthreshold`, `—logappend`, and `-numlogfiles` parameters affect the DataServer's log file when specified as startup parameters to an OpenEdge client or a `probrkr` broker process starting a remote DataServer server process. You can also specify these parameters in the `srvrDSLogFile`, `srvrLogThreshold`, `srvrNumLogFiles` and `srvrLogAppend` properties, respectively, of a DataServer broker instance defined to the Unified Broker and manage server log files for an AppServer agent connected to a DataServer. All the server processes affected by these parameters will write to the same `-dslog or srvrDSLogFile` log file and switch to the next log file when the logthreshold is reached.

All the server processes affected by these parameters will write to the same `-dslog` log file and switch to the next log file when the log threshold is reached.The code page used for the log file is `-cpinternal` and no conversions are performed for these log files.

For more information about the logging settings for the client and server contexts, see the reference entries for the LOG-ENTRY-TYPES attribute and the LOGGING-LEVEL attribute in the *OpenEdge Development: ABL Reference*.

For more information on using `-Dsrv` to specify these logging settings for the server context, see *OpenEdge Development: Debugging and Troubleshooting*, and OpenEdge DataServer guides.

Maximum Dynamic DataServer Port (-dsmaxport)

Use maximum Dynamic DataServer Port (`-dsmaxport`) when starting ProBroker to specify the highest port number in a specified range of port numbers used to start DataServer servers.

Operating system and syntax	UNIX / Windows	**`-dsmaxport` *`n`***		
Use with	**UNIX Maximum value**	**UNIX Minimum value**	**Windows Maximum value**	**Windows Minimum value**
DataServer	2000	1025	5000	3000

`n`

The port number that is the highest in a specified range.

You specify the lowest port number with the Minimum Dynamic DataServer Server (`-dsminport`) parameter. The range of port numbers defined by the Maximum Dynamic DataServer Server (`-dsmaxport`) and Minimum Dynamic DataServer Server (`-port`) parameters provides client access to an OpenEdge server that is behind a firewall. Some operating systems choose transient client ports in the 32,768-to-65,535 range. Choosing a port in this range might produce unwanted results.

Minimum Dynamic DataServer (-dsminport)

Use Minimum Dynamic Server (`-dsminport`) to specify the lowest port number in a specified range of port numbers accessible to a client.

Operating system and syntax	**UNIX / Windows**	**`-dsminport`** ***`n`***		
Use with	**UNIX Maximum value**	**UNIX Minimum value**	**Windows Maximum value**	**Windows Minimum value**
DataServer	2000	1025	5000	3000

`n`

The port number that is the lowest in a specified range.

You specify the highest port number with the Maximum Dynamic DataServer Server (-dsmaxport) parameter. The range of port numbers defined by the Maximum Dynamic DataServer Server (-dsmaxport) and Minimum Dynamic DataServer Server (-dsminport) parameters provides client access to an OpenEdge server that is behind a firewall. Some operating systems choose transient client ports in the 32,768-to-65,535 range. Choosing a port in this range might produce unwanted results.

DataServer (-Dsrv)

Use DataServer (`-Dsrv`) to tell OpenEdge that the specified keywords are parameters for the ODBC, ORACLE, or MS SQL Server DataServer.

Operating system and syntax	**UNIX / Windows**	**`-Dsrv`** ***`keyword`*** **`( ,`** ***`value`*** **`)`** [**`,`** ***`keyword2`*** **`( ,`** ***`value2`*** **`)`** . . .]		
Use with	Maximum value	Minimum value	Single-user default	Multi-user default
DataServer	—	—	—	—

```
keyword ( , value) [ , keyword2 ( , value2 ) ...]
```

Parameters, with optional values, for DataServers.

These parameters are specific to ODBC data sources or are required to connect to certain data sources using the OpenEdge DataServers. You can specify up to 50 keywords with the `-Dsrv` parameter.

There are three types of DataServer parameters:

- OpenEdge DataServer for ODBC parameters
- ODBC SQL connection and statement parameters
- OpenEdge DataServer for ORACLE parameters

For more information about the parameters, see the OpenEdge DataServer guides.

Database Type (-dt)

Use Database Type (`-dt`) to specify one of the following database types: Progress, ODBC, ORACLE, and AS/400.

Operating system and syntax	**UNIX / Windows**	**`-dt` *`db-type`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Connection, DataServer	—	—	Progress	Progress

`db-type`

The database type.

The value is not case sensitive. When connecting to a non-OpenEdge database, use Database Type (`-dt`) to specify the type of database.

European Numeric Format (-E)

Use European Numeric Format (`-E`) parameter to tell the AVM to display thousands separators as periods and decimal points as commas (for example: 1.234.567,89) when displaying or prompting for numeric values.

Operating system and syntax	UNIX / Windows	**`-E`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	AMERICAN	AMERICAN

`AMERICAN` is the default value, in which periods are displayed as decimal points and commas as thousands separators (for example: 1,234,567.89).

When you specify `EUROPEAN`, continue to use periods as decimal points and commas as thousands separators in your source files. Note there is no command line parameter to specify `AMERICAN`, or reset `EUROPEAN`.

Numeric format specifications in the Data Dictionary and the decimal constants in procedures always must use the period to represent the decimal point.

You can also use the `NUMERIC-FORMAT` attribute on the `SESSION` system handle to get or set the numeric format.

Enforce Mm (-enforceMm)

Use Enforce Mm (`–enforceMm`) to specify that the behavior of the Message Buffer Size (`–Mm`) startup parameter follow prior releases.

Operating system and syntax	UNIX / Windows	**`-enforceMm`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Connection, Database Server	—	—	—	—

OpenEdge uses message buffers to move records (messages) between servers and remote clients. The size of the message buffer is specified with `–Mm`. In OpenEdge Release 11.6, the value of `–Mm` is not required to agree between the server and the client, however in releases prior to OpenEdge 11.6, the value of `–Mm` had to be specified the same for both the client and the server or the connection is refused. Use Enforce Mm (`–enforceMm`) to require agreement between the client and server values of `–Mm` as in earlier releases.

Entity Expansion Limit (-entityExpansionLimit)

Use Entity Expansion Limit (`-entityExpansionLimit`) to set an upper limit on how many entity substitutions the XML parser (DOM or SAX) will allow in a document.

Operating system and syntax	**UNIX / Windows**	**`-entityExpansionLimit` *`n`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	2,147,483,647	0	50,000	50,000

`n`

> The number of entity substitutions the XML parser will allow. Valid values are integers equal to or greater than 0. If the parameter is not specified, the default value is 50,000.

By limiting the number of entity substitutions, use of this parameter prevents recursive entities from consuming system resources.

This behavior can also be controlled on a case-by-case basis by the `ENTITY-EXPANSION-LIMIT` attribute of a particular `SAX-reader` or `X-document` object handle or the `XML-ENTITY-EXPANSION-LIMIT` of the `WEB-OBJECT` system handle. Setting the `ENTITY-EXPANSION-LIMIT` or `XML-ENTITY-EXPANSION-LIMIT` attribute overrides the behavior indicated by the startup parameter. See *OpenEdge Development: ABL Reference* for more information.

Error Stack (-errorstack)

Use Error Stack (`-errorstack`) to allow error objects to save the ABL call stack in the `CallStack` property of an error object at the time the AVM generates the error object.

When `-errorstack` is specified, the `ERROR-STACK-TRACE` attribute of the `SESSION` handle is set to `TRUE`. The default value for this attribute is `FALSE`. While saving the call stack is a useful debugging feature, it can also consume resources and can potentially affect performance. Enabling `ERROR-STACK-TRACE` with this startup parameter is not recommended in a production environment.

Operating system and syntax	**Windows**	**`-errorstack`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

Event Level (-evtlevel)

Use Event Level (`-evtlevel`) to specify the level of information that OpenEdge writes to the Windows Application Event Log.

Operating system and syntax	Windows	**`-evtlevel`** ***`value`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session, Database Server	—	—	Normal	Normal

`value`

- **`None`** — No OpenEdge events are written to the Event Log.
- **`Brief`** — OpenEdge Error and Warning messages are written to the Event Log.
- **`Normal`** — OpenEdge Error and Warning messages are written to the Event Log along with any OpenEdge message that is normally written to the log file (`.lg`). This is the default.
- **`Full`** — OpenEdge Error, Warning, and Informational messages are written to the Event Log along with any messages generated by the Message Statement.

For more information about OpenEdge and the Windows Event Log, see *OpenEdge Getting Started: Installation and Configuration*.

Expand Browse (-expandbrow)

Use Expand Browse (`-expandbrow`) to turn on the `EXPANDABLE` and `FIT-LAST-COLUMN` options for all browsers created in the current session.

Operating system and syntax	Windows	**`-expandbrow`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

The `EXPANDABLE` attribute has the same behavior as the `FIT-LAST-COLUMN` attribute. Therefore, if the `-expandbrow` parameter is specified at startup, the `FIT-LAST-COLUMN` attribute is set to `TRUE` for each browse in that session.

Note: Progress Software Corporation recommends that you use the `FIT-LAST-COLUMN` attribute instead of the `EXPANDABLE` attribute. This recommendation includes replacing `EXPANDABLE` with `FIT-LAST-COLUMN` in your current code.

For more information on these attributes, see *OpenEdge Development: ABL Reference*.

Force Access (-F)

Use Force Access (`-F`) with the PROSHUT command to invoke an emergency shutdown of a shared-memory database.

Operating system and syntax	**UNIX / Windows**	**`-F`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Connection	—	—	—	—

Note: The PROMON and PROUTIL utilities also use a `-F` parameter; however, the results are not the same. In addition, using `-F` with PROUTIL might compromise the integrity of the database. For more information, see *OpenEdge Data Management: Database Administration*.

Schema Field Cache Size (-fc)

Use Schema Field Cache Size (`-fc`) to change the number of entries in the schema field cache.

Operating system and syntax	**UNIX / Windows**	**`-fc` *`num-entries`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	0	128	128

`num-entries`

The number of entries in the schema field cache.

For example, you might increase the size if you retrieve large numbers of fields from your database. The schema field cache reduces the time required to compile SQL queries by storing schema field information in memory. For best results, set this parameter to the total number of fields retrieved from the database.

If you set the value to 0, OpenEdge disables schema field caching.

OpenEdge uses approximately 150 additional bytes of memory per schema field cache entry used. Because memory is allocated when required, unused entries produce minimal overhead.

Filtering Asynchronous COM Events (-filterocxevents)

Use Filtering Asynchronous COM Events (`–filterocxevents`) to control the conditions under which you want asynchronous COM events handled.

Operating system and syntax	UNIX / Windows	**-filterocxevents**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

An asynchronous COM event is one that is sent by an ActiveX control or Automation object because of an external event, rather than in response to user input. One example of this is the Tick event from the PSTimer. Another is an event signalling the receipt of a message from an external source, such as a control that implements a mail notification system.

By default, asynchronous COM events are handled when any of the following conditions occur:

- While an application is waiting for user input, such as during a `WAIT–FOR` or `UPDATE` statement
- While an alert box from a `MESSAGE` statement is on the screen
- During trigger processing, if the `MultitaskingInterval` parameter is set to a number greater than 0

Handling COM events during the second and third conditions might cause random problems in your application, depending on what ABL code is executed in the event trigger. You use the `–filterocxevents` startup parameter to prevent asynchronous COM events from being handled during the second and third conditions.

Field List Disable (-fldisable)

Use Field List Disable (`-fldisable`), which is a run-time parameter, to cause the AVM to ignore field lists in the r-code and fetch complete records.

Operating system and syntax	UNIX / Windows	**-fldisable**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

Also, note the difference between `-fldisable` and `-rereadfields`. The Field List Disable (`-fldisable`) option causes the AVM to ignore **all** field lists and to fetch the entire record for every query. The Reread Field List (`-rereadfields`) option causes the AVM to ignore a field list and fetch the entire record only when an error occurs due to a missing field. Therefore, using `-rereadfields` is likely to have less of a negative impact on performance.

See Reread Fields (-rereadfields) on page 187 for more information. For more information about using field lists, see *OpenEdge Getting Started: ABL Essentials*.

Before-image Truncate Interval (-G)

Use Before-image Truncate Interval (`-G`) to specify the number of seconds before OpenEdge reuses a before-image cluster.

Operating system and syntax	**UNIX / Windows**	**`-G n`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	—	0	0	0

n

The number of seconds OpenEdge waits.

Note: This parameter was previously known as Before-image Cluster Age.

Group Delay (-groupdelay)

Use Group Delay (`-groupdelay`) to increase performance when Delayed BI File Write (`-Mf`) is set to zero.

Operating system and syntax	**UNIX / Windows**	**`-groupdelay n`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	1,000	0	0	1

n

The number of milliseconds a transaction waits before committing.

When the Group Delay is set greater than zero (0), OpenEdge uses a technique known as Group Commit. When using Group Commit, a transaction spools its end note to the BI buffer and waits a short time until the buffer becomes full and is written to disk, or waits for other transactions to end and store their end notes in the BI buffer so that several transactions are committed by the same synchronous write. In this manner, Group Commit benefits overall performance, although each individual transaction might take slightly longer.

Host Name (-H)

Use Host Name (`-H`) to identify the host name.

Operating system and syntax	**UNIX / Windows(TCP)**	**`-H {`*`host-name`*`	localhost`[11]`}`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Connection, Database Server, DataServer	—	—	—	—

`host-name`

The name (address) of the database server machine. This name is assigned to the machine in your TCP/IP hosts file.

`localhost`

A reserved word that specifies that the database server communicate only with clients on the database server machine.

Number of Databases (-h)

Use Number of Databases (`-h`) to connect to more than five databases during a single OpenEdge session.

Operating system and syntax	**UNIX / Windows**	**`-h`** ***`n`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	240	1	5	5

[11] **`Localhost` does not apply to DataServers.**

n

The maximum number of databases that can be connected during an OpenEdge session.

OpenEdge automatically allocates enough space to allow the connection of five databases.

Hardlimit (-hardlimit)

Use Hardlimit (`-hardlimit`) to force OpenEdge to adhere to the set limits for Directory Size (-D) on page 100, Local Buffer Size (-l) on page 129, Maximum Memory (-mmax) on page 150, and Nested Blocks (-nb) on page 157.

Operating system and syntax	UNIX / Windows	**-hardlimit**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

Use Hardlimit (`-hardlimit`), Directory Size (-D) on page 100, Local Buffer Size (-l) on page 129, Maximum Memory (-mmax) on page 150, and Nested Blocks (-nb) on page 157 are by default soft limits; that is, OpenEdge tries to manage system resources within the limits given, but when it cannot, OpenEdge increases the exceeded limit.

When -hardlimit is specified and one of the limits is exceeded, OpenEdge generates a `WARNING` message that is written to the current output (if there is current output) and to the log file (`LG`). In addition, OpenEdge issues a message that a resource limit was reached and raises an untrappable `STOP` condition.

Hash Table Entries (-hash)

Use Hash Table Entries (`-hash`) to specify the number of hash table entries to use for a buffer pool.

Operating system and syntax	UNIX / Windows	**-hash *n***	
Use with	**Maximum value**	**Minimum value**	**Default**
Client Connection, Database Server	—	13	Approximately 1/4 of the -B value

n

The number of hash table entries to use for the buffer pool.

Caution: Do not use this parameter unless directed to do so by Progress Software Corporation Technical Support.

No Crash Protection (-i)

Use No Crash Protection (`-i`) to tell OpenEdge to run without using database integrity or database recovery.

Operating system and syntax	**UNIX / Windows**	**`-i`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Connection, Database Server	—	—	—	—

When running OpenEdge without database integrity, it writes fewer data and before-image blocks to the disk. In this mode, some ABL procedures (such as those that create and delete large numbers of records) run significantly faster than if they are running with database integrity.

When running OpenEdge with the `-i` parameter, transaction undo is supported. Therefore, there will still be a before-image file, which might grow quite large during very long transactions.

Use this parameter to do bulk data loading or for large batch runs. It reduces the number of disk input or output operations. Loading a database for the first time is a good example of a use for this parameter.

Caution: If you run OpenEdge with the `-i` parameter and OpenEdge fails for any reason, you cannot recover the database.

Do not use the `-i` parameter unless you have a complete backup of the database and can rerun procedures in case of a system failure. If the system fails during an OpenEdge session started without crash protection, restore the backup copy and rerun the necessary procedures. For information about restoring a database, see *OpenEdge Data Management: Database Administration*.

The following messages might appear when starting an OpenEdge session after a system failure:

- After a system failure while a full-integrity OpenEdge session was running:

```
**The last session was abnormally terminated.
**Any incomplete transactions are being backed out.
**Database recovery is complete. You must rerun all active transactions.
```

- After a system failure while a no-integrity OpenEdge session was running:

```
**The last session was abnormally terminated.
**The last session was run with the -i no integrity option.
**Your database cannot be repaired, you must restore a backup copy.
```

Dynamics Parameter (-icfparam)

Use Dynamics Parameter (`-icfparam`) to specify a character string that can be accessed from ABL procedures within the Progress Dynamics framework.

Operating system and syntax	**UNIX / Windows**	**`-icfparam`** ***`string`***		
Use with	Maximum value	Minimum value	Single-user default	Multi-user default
Client Session	—	—	—	—

`string`

A character string that supplies dynamics procedures with Dynamics-related data.

You can access this string within an ABL procedure by reading the `ICFPARAMETER` attribute of the `SESSION` system handle.

Note: This parameter is reserved for use by Dynamics and procedures that have been integrated with Dynamics. Using this parameter for any purpose other than operating within the Dynamics framework will interfere with your ability to integrate your application with that framework at a later time.

For more information about the `ICFPARAMETER` attribute or the `SESSION` system handle, see *OpenEdge Development: ABL Reference.*

Index consistency check (-IndexCheck)

Use Index consistency check to enable consistency checking for all index blocks in the specified index or index partition.

Operating system and syntax	UNIX / Windows	`-IndexCheck {`
		`tablename.indexname` |
		`ownername.tablename.indexname` |
		`ownername.tablename.indexname.partitionname`
		`}`		

Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Connection, Database Server	—	—	—	—

`ownername`

> Specifies the owner of the table containing the index you want to check. You must specify an owner name unless the table's name is unique within the database, or the table is owned by PUB. By default, ABL tables are owned by PUB.

`tablename`

> Specifies the table containing the index to be checked.

`indexname`

> Specifies the name of the index for consistency checking.

`partitionname`

> Specifies the partition of the index for consistency checking.

When enabled, this option applies consistency checks to all index blocks and index write operations.

Index consistency check validates a block is still physically correct after an operation has been performed. For example, after an index key is inserted into an index block, the consistency check validates that the block still laid out correctly.

You can only specify one index or index partition with `-IndexCheck`. To check multiple index partitions, apply consistency checking to the entire index. To check multiple indexes, you can apply consistency checking to the entire database with Database consistency check (-DbCheck) on page 103, or if multiple indexes are in one area, you can apply consistency checking to an entire area with Area consistency check (-AreaCheck) on page 70.

When specifying the index or index partition to check, the following parsing rules are applied:

- If two variables are supplied, they are interpreted as the table name and the index name (*`tablename.indexname`*).

- If three variables are supplied, they are interpreted as the ownername, the table name, and the index name (*ownername.tablename.indexname*).
- Four variables **must** be supplied to specify a partition of an index (*ownername.tablename.indexname.partitionname*).

Index consistency checking can be enabled or disabled while your database is online with PROMON. See *OpenEdge Data Management: Database Administration* for more information.

Index consistency check can be enabled for a single user client or for any offline utility. If the utility is read-only, such as DBANALYS, the `-IndexCheck` parameter is ignored. Online utilities determine whether or not to perform index consistency checking during execution based on the use of `-IndexCheck` at broker startup or by the enablement/disablement of the index consistency checking in PROMON.

Index Range Size (-indexrangesize)

Use Index Range Size (`-indexrangesiz`e) to specify the number of indexes for which you want to collect statistics.

Operating system and syntax	**UNIX / Windows**	**`-indexrangesize`** ***`n`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	—	—	—	—

`n`

The number of indexes for which you want to track access statistics.

For more information about tracking index statistics, see *OpenEdge Data Management: Database Administration*.

Initialization File (-ininame)

Use Initialization File (`-ininam`e) to specify the location, in the registry, for the application's initialization information.

Operating system and syntax	**Windows**	**`-ininame`** ***`name`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

name

Name of the initialization registry subkey or the application's initialization (.INI) file.

When the Registry Base Key (-basekey) on page 76 parameter is not specified at startup, the AVM searches the `HKEY_CURRENT_USER` key followed by the `HKEY_LOCAL_MACHINE` key for the specified name. If no registry key of the specified name is found, the AVM searches for a `.INI` file that matches *name*.

You can only use this startup parameter at the command line. It is ignored when used in a `.pf` file.

Note: The `-ininame` parameter replaces the `-name` option for the Windows Attributes (`-Wa`) parameter for Version 7.3B and later.

Input Characters (-inp)

Use Input Characters (`-inp`) to expand the available buffer space for a single ABL statement.

Operating system and syntax	**UNIX / Windows**	**-inp *n***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	2147483647	1	15000	15000

n

The number of characters allowed in a single ABL statement. The default is 15000 characters.

If not specified and you exceed 15000 characters in a single statement, the AVM displays a message indicating that the statement is too long and advises you to increase the number of input characters per statement using this parameter.

Internet Protocol (-ipver)

Use Internet Protocol (`-ipver`) to identify the IP version for network connections.

Operating system and syntax	**UNIX / Windows**	**`-ipver { IPv4 | IPv6 }`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session, Database Server	—	—	—	`IPv4`

`IPv4`

Specifies Internet Protocol Version 4. Only network connections with IPv4 are allowed.

`IPv6`

Specifies Internet Protocol Version 6. IPv6 allows network connections with IPv6 addresses and mapped IPv4 addresses. OpenEdge supports V4 mapped addresses where underlying operating system support exists: UNIX, Linux, and Windows Vista.

Note: The `-ipver` startup parameter is case sensitive and must be specified in all lower case. The values `IPv4` and `IPv6` are not case sensitive, and can be specified in any case.

Initial Value Segment No Convert (-isnoconv)

Use Initial Value Segment No Convert (`-isnoconv`) to disable a code page conversion that Version 9.1A provides and that previous versions do not.

Operating system and syntax	**UNIX / Windows**	**`-isnoconv`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

Consider using `-isnoconv` when all of the following conditions occur:

- An ABL program contains character variables initialized from text literals.
- The text literals contain international characters (whether or not the literal is flagged as untranslatable using the `:U` attribute).

- The code page of the r-code (`-cprcodein`) does not match the code page of internal memory (`-cpinternal`).
- This is a compile-time option, not a run-time option, and affects the generation of r-code, not its execution.

When you run earlier versions of OpenEdge under these conditions, and the AVM reads text literals from r-code into internal memory, it does not convert the code page. To compensate for this, programmers often pre-convert international characters in text literals to `-cpinternal`.

When you run Version 9.1A under these conditions, and the AVM reads text literals from r-code into internal memory, it converts the code page from `-cprcodein` to `-cpinternal`. This code page conversion might cause international characters you have pre-converted to -`cpinternal` to be converted again to `-cpinternal`, which garbles them.

Note: Progress Software Corporation recommends that you avoid preconverting text literals.

Keyword Forget List (-k)

Use Keyword Forget List (`-k`) to disable ABL keywords. The keyword forget list is provided to ease migration from one release of OpenEdge to the next.

Operating system and syntax	**UNIX / Windows**	**-k *filename***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

`filename`

The name of a file that contains ABL keywords to disable. Each keyword must use a single line in the file. If the AVM encounters a word that is not an ABL keyword, it returns a warning message but continues processing the file. OpenEdge uses the `PROPATH` variable to locate *`filename`*. If it cannot find *`filename`*, it displays an error message and does not start.

Do not use keywords in procedures as ABL user-defined element names (tables, fields, frames, variables, streams, and so on). If necessary, disable new keywords (and the features they implement) until they can be removed from the application.

The directory where OpenEdge is installed contains the following keyword files:

- **`newkywd`** — Lists all keywords that are new in the most recent release of OpenEdge
- **`newkw810`** — Use instead of `newkywd`, if you are converting from Progress Version 8 to OpenEdge Release 10
- **`newkw710`** — Use instead of `newkywd`, if you are converting from Progress Version 7 to OpenEdge Release 10
- **`newkw610`** — Use instead of `newkywd`, if you are converting from Progress Version 6 to OpenEdge Release 10

Key Alias (-keyalias)

Use Key Alias (`-keyalias`) to identify a SSL private key/digital certificate key-store other than the default.

Operating system and syntax	**UNIX / Windows**	**`-keyalias` *`key-alias-name`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	—	—	—	default_server

`key-alias-name`

Specifies the alias name of the Secure Sockets Layer (SSL) private key/digital certificate key-store entry to use.

Key Alias Password (-keyaliaspasswd)

Use Key Alias Password (`-keyaliaspasswd`) to allow access to the Key Alias when you use a Key Alias other than the default.

Operating system and syntax	**UNIX / Windows**	**`-keyaliaspasswd` *`key-alias-password`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	—	—	—	password[12]

`key-alias-password`

Specifies the encrypted Secure Sockets Layer (SSL) Key Alias Password to use to access the server's private key/digital certificate key-store entry. The default is `password`.

The *`key-alias-password`* value must be encrypted. You can use the `genpassword` utility, located in your installation's `bin` directory, to encrypt the password.

Lock Table Entries (-L)

Use Lock Table Entries (`-L`) to change the limits of the record locking table.

[12] The actual value used is the encrypted value of the string "password."

Note: Starting in Release 10.1C, the OpenEdge RDBMS expands the range of internal validations used to ensure database consistency in both index and data blocks during forward processing. Validations using PROUTIL can be run online as part of routine health checks. For more information, see *OpenEdge Data Management: Database Administration*.

Operating system and syntax	**UNIX / Windows**	**`-L n`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	System dependent[13]	32	—	8192

n

The number of entries in the record locking table. If you specify a value that is not a multiple of 32, OpenEdge rounds the value you specify to the next highest multiple of 32.

Each record that is accessed by any user takes one entry. This is true whether the record is accessed with `SHARE-LOCK` or `EXCLUSIVE-LOCK`.

Increase the size of the lock table if the following message appears:

```
SYSTEM ERROR: Record lock table too small. Increase -L parameter.
```

This message also might indicate that a particular procedure should be restructured into smaller transactions or should be run in single-user rather than multi-user mode. When lock table limits are exceeded, check to make sure transactions are not too large before increasing the lock table size.

If a user process tries to acquire a lock and the lock table overflows, the user's program is aborted, but the server continues to operate. Any partial transactions are undone.

Note: The two record locks are acquired when records are accessed with the `BREAK BY` option (in `DO`, `FOR EACH`, or `REPEAT` statements). Each lock table entry takes 18 bytes on typical systems.

[13] Limited by available memory.

Local Buffer Size (-l)

Use Local Buffer Size (`-l`) to change the size of the local record buffer in which the AVM stores all variables, work files, and records that are in use at one time for a user.

Operating system and syntax	**UNIX / Windows**	**`-l` *`n`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	4,000,000,000	1	200	200

`n`

The size of the local record buffer in 1KB units.

The maximum size of the local record buffer is 4GB for most systems. If you get one of the following error messages, you must increase the size of this buffer:

```
SYSTEM ERROR: bfget: No space. Increase -l parameter.
SYSTEM ERROR: bfxpnd: No space. Increase -l parameter.
```

The Local Buffer Size (`-l`) limit is a soft limit; so if your application needs to exceed the limit, OpenEdge automatically increases it and issues a `WARNING` message. The `WARNING` is written to the current output (if there is current output) and to the log file (`LG`).

You can force OpenEdge to adhere to the specified `-l` limit by starting the session with the Hardlimit (-hardlimit) on page 119 startup parameter. When you use the `-hardlimit` startup parameter, OpenEdge issues the `WARNING` message when you exceed the Local Buffer Size limit. It also issues a message that a resource limit was reached and raises an untrappable `STOP` condition. Note that -hardlimit also enforces the limits set by the Directory Size (-D) on page 100, the Maximum Memory (-mmax) on page 150, and the Nested Blocks (-nb) on page 157 startup parameters, so the `STOP` condition is raised when any of the specified limits is reached.

Logical Database Name (-ld)

Use Logical Database Name (`-ld`) to assign the logical name to a database. If you omit this parameter, OpenEdge uses the physical database name (without the path or `.db` suffix).

Operating system and syntax	**UNIX / Windows**	**`-ld`** ***`logical-dbname`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Connection	—	—	—	—

`logical-dbname`

The logical database name.

The name must be a valid identifier, which means:

- It can be up to 32 characters long.
- It can be any combination of English letters and numbers, underscore (_), and dash (-) characters.
- The first character must be an uppercase or lowercase letter.

A logical database name cannot include ABL reserved words, a space, any accented letters, or any of the following special characters:

```
\ "  '  *  ;  |  ?  [  ]  (  )   !   {  }  <  >
```

For more information on logical database names, see *OpenEdge Getting Started: ABL Essentials* and *OpenEdge Data Management: Database Administration*.

Lock Governor (-LGovernor)

Use Lock Governor (`-LGovernor`) to specify the percentage of the lock table that a tenant can occupy in a multi-tenant database. The valid value for the Lock Governor ranges from 1 to100.

Operating system and syntax	**UNIX / Windows**	**`-LGovernor`** ***`n`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	100	1	—	100

n

The percent of the lock table one tenant can occupy.

Note: Lock governor applies to the default tenant and a regular tenant. This governor does not apply to super-tenants and non multi-tenant databases.

Lock governor prevents a tenant from acquiring all the locks in a table. When a tenant attempts to acquire more locks than the governor would allow, an error message is displayed and recorded in the database log as:

```
Attempt to exceed tenant lock governor for tenant <tenantID>. Increase -L or
 -LGovernor.
```

The Startup file Virtual System Table (VST) contains field for the Lock Governor startup parameter. For more information on Startup file, see *OpenEdge Data Management: Database Administration*.

The PROMON utility helps you to monitor and change Lock Governor values. For more information on how the PROMON utility monitors online Lock Governor, see *OpenEdge Data Management: Database Administration*.

Literal Question (-literalquestion)

Use Literal Question (`-literalquestion`) to change the default value of the `LITERAL-QUESTION` attribute to `TRUE` (which would otherwise be `FALSE`).

Operating system and syntax	UNIX / Windows	**`-literalquestion`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

The `LITERAL-QUESTION` attribute lets you specify how the AVM interprets a quoted character value during assignment into the `BUFFER-VALUE` attribute for a character `BUFFER-FIELD` object. That is, whether the AVM treats the quoted character value as a literal or non-literal character value.

When `TRUE`, the **new** default value, the AVM treats a quoted character value as a literal character value. That is, it does not remove enclosing quotes, trailing blanks, or formatting insertion characters.

When `FALSE`, the AVM treats a quoted character value as a non-literal character value. That is, it removes enclosing quotes, trailing blanks, and formatting insertion characters. For example:

- The AVM treats "`abc     `" as "`abc`".
- The AVM treats a quoted question mark character ("?") as the `Unknown value (?)`.

For more information about ABL objects and attributes, see *OpenEdge Development: ABL Reference*.

Lock Table Hash Size (-lkhash)

Use Lock Table Hash Size (-lkhash) to specify the size of the hash table that controls access to the lock table.

Operating system and syntax	UNIX / Windows	**-lkhash *n***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	—	—	—	—[14]

Progress Software Corporation recommends changing the value of Lock Table Hash Size (-lkhash) only after contacting Technical Support.

Lock release (-lkrela)

Lock Release (-lkrela) uses the original lock release mechanism installed from previous OpenEdge versions.

Operating system and syntax	UNIX / Windows	**-lkrela**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

Using this parameter instructs the database to use the original lock release mechanism installed in previous versions of OpenEdge (versions 10.1A and higher).

Note: When using this parameter, keep in mind that the database engine may search all lock chains to be released at transaction end and disconnect time. As a result, you may see a degradation in performance.

[14] The default is variable based on the size of your lock table (-l).

Lock Timeout (-lkwtmo)

Use Lock Timeout (`-lkwtmo`) to specify a different wait time (in seconds).

Operating system and syntax	UNIX / Windows	**-lkwtmo** ***seconds***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	10	1800[15]	1800[15]

seconds

The wait time, in seconds.

An OpenEdge process encountering a locked resource waits for a limited time before continuing execution. If the resource is still locked, the process times out and the wait is canceled. This feature is called "lock wait timeout."

The default wait time is 1800 seconds (30 minutes) for a single-user client session as well as a multiple-user client session. The minimum value is 10 seconds.

Note: The wait time is accurate to within one minute of the value specified.

Lock timeout applies to interactive and batch clients. The client gets a `STOP` condition if this value is exceeded on the AppServer. For more information, see the sections on client/server conflicts in *OpenEdge Application Server: Developing AppServer Applications*.

Lock timeout affects `WAIT-FOR` record locks, schema locks, and transaction commit locks. Lock timeout does not affect internal locks used inside the database manager, such as buffer locks.

Language (-lng)

Use Language (`-lng`) to specify the initial return value for the `CURRENT-LANGUAGE` function. This setting determines from which r-code segment the AVM reads character-string constants.

Operating system and syntax	UNIX / Windows	**-lng** ***language-name***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	?	?

[15] The default wait time for WebSpeed is 10 seconds.

`language-name`

A character string that contains the name of the current language.

The default value for the Language parameter is a character string that contains the question mark (`?`) character, which is not to be confused with the `Unknown value (?)`. This tells the AVM to use the default language for the application.

For more information on the Language startup parameter, see *OpenEdge Development: Internationalizing Applications* and *OpenEdge Development: ABL Reference*.

Log Entry Types (-logentrytypes)

Use Log Entry Types (`-logentrytypes`) to specify one or more types of log entries to write to the log file specified by the Client Logging (-clientlog) on page 86 and the DataServer Logging (-dslog) on page 108 startup parameter.

Operating system and syntax	**UNIX / Windows**	**`-logentrytypes`** ***`string`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

`string`

A character string that specifies a comma-separated list of log entry types.

By default, the logging level you specify using the Logging Level (-logginglevel) on page 139 startup parameter applies to all log entry types specified. However, you can specify a different logging level for each entry type, as follows:

```
log-entry-type [ :level ]
```

`log-entry-type`

A log entry type listed in the tables below.

`level`

A logging level value (between 0 and 4).

The following table describes the log entry types for the client context.

Table 2: Log entry types (Client Context)

Log entry type	Executable	Description
`4GLMessages`	ABL (GUI and character mode). AppServer and WebSpeed do not require this log entry type for ABL messages to be written to the log file	Turns on logging of ABL messages. OpenEdge writes all ABL `VIEW-AS ALERT-BOX` messages to the log file, together with the ABL stack, when you turn on Debug Alert, using either the Debug Alert (-debugalert) on page 104 startup parameter or the `DEBUG-ALERT` attribute on the `SESSION` system handle.
`4GLTrace`	ABL clients, AppServer, and WebSpeed agents	Turns on logging for the execution of internal procedures, user-defined functions, persistent user-interface triggers, and named events (generated by the `RUN`, `FUNCTION`, `PUBLISH`, and `SUBSCRIBE` statements, respectively). It also logs the instantiation and use of classes, including execution of constructors (invoked by the `NEW` phrase and also by the `SUPER` and `THIS-OBJECT` statements), the execution of methods defined within classes (including those invoked using the `SUPER` system reference), the execution of property accessor methods (invoked by accessing a property of a class), and the execution of destructors (invoked by executing the `DELETE OBJECT` statement).
`4GLTrans`	ABL clients	Turns on logging for the processing of transactions and subtransactions in ABL procedures.

`AiaMgmt`
`AiaProp`
`AiaRqst`
`AiaUbroker`
`AiaDefault`
	AIA	Turns on logging for the AIA component.

`ASDefault`	AppServer agent	Combines the `ASPlumbing` and `DB.Connects` log entry types. It is the default value for AppServer agents.

Log entry type	Executable	Description
ASPlumbing	AppServer agent	Turns on logging for different actions, depending on the logging level specified.
DB.Connects	ABL clients, AppServer, and WebSpeed agents	Turns on logging of database connections (connects and disconnects). The log messages include database name and user ID number.
DS.Cursor	DataServer clients	Turns on logging for DataServer client-side processing for cursors used to position the OpenEdge client against the foreign data source.
DS.QryInfo	DataServer clients	Turns on logging for DataServer client-side processing for queries sent to the foreign data source.
DynObjects.DB DynObjects.XML DynObjects.Other DynObjects.Class DynObjects.UI	ABL clients, AppServer, and WebSpeed agentsABL clients and WebSpeed agents	Turns on logging of dynamic object creation and deletion.
FileID	ABL clients, AppServer, and WebSpeed agents	Turns on logging of file access operations (opening and closing files), and any file access error messages that might occur.
IgnoredOps	ABL clients	Turns on logging of operations that the AVM ignores because the ABL windows to which they apply are embedded in .NET forms. For information about operations that the AVM ignores when they occur in the context of an embedded window, see the EmbeddedWindow property reference entry in *OpenEdge Development: ABL Reference*.
MsgrTrace	WebSpeed Messengers	Turns on logging for WebSpeed Messengers. The information logged depends on which Messenger is running and the logging level specified.

Log entry type	Executable	Description
`NSPlumbing`	NameServer	Turns on logging for the NameServer component.

`ProEvents.UI.Char`
`ProEvents.UI.Command`
`ProEvents.Other`
	ABL clients, AppServer, and WebSpeed agents	Turns on event logging for different categories of events.

`QryInfo`	ABL clients, AppServer, and WebSpeed agents	Turns on logging of queries (each open query and `FOR EACH` block) executed in an application.
`SAX`	ABL clients, AppServer, and WebSpeed agents	Turns on logging for the SAX parser.
`Temp-tables`	ABL clients, AppServer servers, WebSpeed agents, and interactive and batch ABL clients	Enables specific logging for temp-tables so that application developers and technical support can gather more information about temp-tables used by an application.
`TTStats`	ABL clients, AppServer, and WebSpeed agents	Enables logging of temp-table statistics to the corresponding server or log files.

`UBroker.Basic`
`UBroker.ClientFSM`
`UBroker.ServerFSM`
`UBroker.ClientMsgStream`
`UBroker.ServerMsgStream`
`UBroker.ClientMsgQueue`
`UBroker.ServerMsgQueue`
`UBroker.ClientMemTrace`
`UBroker.ServerMemTrace`
`UBroker.ThreadPool`
`UBroker.Stats`
`UBroker.AutoTrim`
`UBroker.All`
	Unified Broker	Turns on logging for the Unified Broker component.

`WSADefault`	Web Services Adapter (WSA)	Turns on logging for the Web Services Adapter component.
`DS.Performance`	DataServer clients	Turns on logging for the client-side performance details.

The following table describes the log entry types for the server context.

Table 3: Log entry types (Server Context)

Log Entry Type	Executable	Description
`SQL`	DataServer broker	Turns on logging of SQL query processing in the DataServer log file.
`Cursor`	DataServer broker	Turns on logging of cursor details in the DataServer log file.
`Trans`	DataServer broker	Turns on logging for the processing of transactions and subtransactions in the DataServer log file.
`Connects`	DataServer broker	Turns on logging for connection attributes and related information in the DataServer log file.
`Performance`	DataServer broker	Turns on logging of performance related information in the server context.

For more information about log entry types, see *OpenEdge Development: Debugging and Troubleshooting*.

The following example shows how to specify one or more individual log entry types:

```
-logentrytypes DB.Connects,4GLTrace:2,DynObjects.UI:3
```

The following example shows how to specify all log entry types within a category:

```
-logentrytypes DynObjects.*
```

For more information about logging levels and log entry types, see *OpenEdge Development: Debugging and Troubleshooting*.

You also can turn on logging at run-time by using the LOG-ENTRY-TYPES attribute on the `LOG-MANAGER` system handle. For example, you can include the following line in your ABL code:

```
LOG-MANAGER:LOG-ENTRY-TYPES = "DynObjects.UI"
```

You can turn off logging in your ABL code by setting this attribute to the `Unknown value (?).` For example:

```
LOG-MANAGER:LOG-ENTRY-TYPES = ?
```

For more information about the LOG-ENTRY-TYPES attribute or the `LOG-MANAGER` system handle, see *OpenEdge Development: ABL Reference*.

Logging Level (-logginglevel)

Use Logging Level (`-logginglevel`) to specify the level at which log entries are written to the log file specified by the and the startup parameter. Each logging level specifies a different degree of detail.

Operating system and syntax	**UNIX / Windows**	**`-logginglevel` *`n`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

`n`

A logging level value (between 0 and 4).

Logging level represents the degree of detail written into the log file, as shown in the following table.

Table 4: Logging levels

Logging level	Description
0 (None)	Log no entries. This is equivalent to turning logging off.
1 (Errors)	Log OpenEdge error messages. This includes all error messages and is unrelated to the entry types specified. Errors continue to be logged at all higher levels.
2 (Basic)	Log entry types determine the logged information. Each entry type will generate at least some output. This is the default.
3 (Verbose)	Log entry types determine the logged information.
4 (Extended)	Log entry types determine the logged information.

By default, the logging level you specify applies to all log entry types. However, you can specify individual log entry types with a different logging level using the Log Entry Types (-logentrytypes) on page 134 startup parameter. The higher logging levels include the log information from the lower logging levels, in addition to the information logged at that level.

You can also use attributes on the `LOG-MANAGER` and the `DSLOG-MANAGER` system handle to specify log entry types and logging levels in the client and server contexts, respectively. For more information, see *OpenEdge Development: ABL Reference*.

For more detailed information about enabling logging, see *OpenEdge Development: Debugging and Troubleshooting*.

Log Threshold (-logthreshold)

Use Log Threshold (`-logthreshold`) to specify the file size threshold of log files.

Operating system and syntax	UNIX / Windows	**`-logthreshold`** ***`n`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

`n`

A log file size threshold in bytes.

When the current log file becomes equal to or greater than the specified size, OpenEdge renames and saves the log file and creates a new log file. Valid values are:

- **0** — This means there is no limit other than what the operating system imposes. Specify 0 to ignore the Number of Log Files to Keep (-numlogfiles) on page 169 startup parameter setting. This is the default.
- **Between 500,000 and 2,147,483,647** — Values are in bytes (one byte typically holds one character). You can specify a file size up to 2GB, inclusive, but not lower than 500,000.

  If you specify a number lower than 500,000, a run-time error occurs and the OpenEdge process terminates.

OpenEdge names log files based on a sequence number using the following format:

```
<filename>.999999.<extension>
```

For example, if you specify a log file named `my.log`, OpenEdge renames the log file to `my.000001.log` before creating a new log file.

Use the Client Logging (-clientlog) on page 86 startup parameter to specify a log file name for ABL clients.

Note: When a non-zero value is specified for `-logthreshold`, only one client process at a time can open the file specified by the Client Logging (`-clientlog`) startup parameter. In this case it is recommended that each client session specify a different file name.

Use the DataServer Logging (-dslog) on page 108 startup parameter to specify a log file name for the server context of a DataServer client or agent. Use the Number of Log Files to Keep (-numlogfiles) on page 169 startup parameter to specify the number of log files to keep.

You can also can use the `LOG-THRESHOLD` attribute on the `LOG-MANAGER` system handle to specify the file size threshold of log files. For more information, see *OpenEdge Development: ABL Reference*.

LRU force skips (-lruskips)

Use LRU force skips (`-lruskips`) to specify the number of times a buffer in the buffer pool is accessed before it is placed on the Most Recently Used (MRU) end of the LRU chain.

Operating system and syntax	**UNIX / Windows**	**`-lruskips` *`n`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	2GB	0	—	0

`n`

The number of times to access a buffer before moving it to the end of the LRU chain.

Specifying the LRU force skip parameter is helpful when high LRU latch wait timeouts are occurring (observable in PROMON or by VST). Accessing a buffer a specified number of times eliminates the need to acquire the LRU latch each time a buffer is accessed. When the LRU skip value is tuned to your environment, contention on the LRU latch decreases, improving concurrency and performance, and there is no increase in the number of page misses in the buffer pool. If increasing the LRU skip value causes increased buffer pool misses, then the value should be tuned down or turned off altogether. Setting `-lruskips` to zero disables the feature, and is the default.

Note: This parameter is only available for databases with an Enterprise license. LRU Force skips is not implemented for private buffer pool data access.

For more information on buffer pool management, see *OpenEdge Data Management: Database Administration*.

LRU alternate buffer pool force skips (-lru2skips)

Use LRU alternate buffer pool force skips (`-lru2skips`) to specify the number of times a buffer in the alternate buffer pool is accessed before it is placed on the Most Recently Used (MRU) end of the LRU chain.

Operating system and syntax	**UNIX / Windows**	**`-lru2skips` *`n`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	2GB	0	—	0

n

The number of times to access a buffer before moving it to the end of the LRU chain.

Specifying the LRU alternate buffer pool skip parameter is helpful when high LRU latch wait timeouts are occurring (observable in PROMON or by VST) in the alternate buffer pool. Accessing a buffer a specified number of times eliminates the need to acquire the LRU latch each time a buffer is accessed. When the LRU alternate buffer pool skip value is tuned to your environment, contention on the LRU latch decreases, improving concurrency and performance, and there is no increase in the number of page misses in the buffer pool. If increasing the LRU skip value causes increased buffer pool misses, then the value should be tuned down or turned off altogether. Setting `-lru2skips` to zero disables the feature, and is the default.

Note: This parameter is only available for databases with an Enterprise license. LRU force skips is not implemented for private buffer pool data access.

For more information on buffer pool management, see *OpenEdge Data Management: Database Administration*.

Auto Server (-m1)

Use Auto Server (`-m1`) to start an auto server.

Operating system and syntax	UNIX / Windows	`-m1`		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	—	—	—	—

The OpenEdge broker uses the auto server internally to start a remote user server. This is the default. You will never have to use this parameter directly.

Manual Server (-m2)

Use Manual Server (`-m2`) to manually start a remote user server after you start a broker (servers are generally started automatically by the broker process).

Operating system and syntax	UNIX / Windows	`-m2`		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	—	—	—	—

Use this parameter in the following cases:

- For debugging purposes, to start servers directly and observe their behavior
- On systems where automatic server generation is not possible

Secondary Login Broker (-m3)

Use Secondary Login Broker (`-m3`) to start each secondary broker in a network environment where more than one broker is using the same protocol.

Operating system and syntax	**UNIX / Windows**	**`-m3`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	—	—	—	—

The secondary broker logs in clients and starts remote user servers.

Maximum Clients Per Server (-Ma)

Use Maximum Clients per Server (`-Ma`) to specify the maximum number of remote users per database server.

Operating system and syntax	**UNIX / Windows**	**`-Ma` *`n`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	2048	1	—	5 users/server

`n`

The maximum number of remote users per database server. The default is the parameter value, divided by the Maximum Servers (-Mn) on page 151 parameter value.

The Maximum Clients Per Server (-Ma), Minimum Clients per Server (-Mi) on page 148, and Maximum Servers (-Mn) on page 151 startup parameters pertain only to versions of OpenEdge that use shared memory. In addition, these parameters apply only to databases that are accessed from remote network nodes.

In most cases, the default behavior is desirable. Note that the default calculation is usually high because it assumes that all users are remote users, while the number specified with -n includes local users. If servers become overloaded with clients, reset the `-Mn` parameter to increase the number of servers. If experience proves that a given number of remote clients overloads a server or exhausts the file descriptors on the system, set the Maximum Clients Per Server (-Ma) parameter to limit clients per server below that level. A *file descriptor* is an object UNIX uses to identify a particular file. The file descriptors might be exhausted because OpenEdge uses them to identify sockets.

Maximum Area Number (-maxAreas)

Use Maximum Area Number (`-maxAreas`) to specify the highest area number available.

Operating system and syntax	**UNIX / Windows**	**`-maxAreas` *n***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	32000	Variable[16]	32000	32000

n

The maximum area number.

Maximum Area Number specifies the highest area number available for use during the time the database is online. Increase the highest area number by shutting down and restarting the database with a higher value, or remove the `-maxAreas` parameter to have the maximum number available.

At database startup, the maximum area number specified with `-maxAreas` is compared to the areas defined in the database. If any defined area has a higher area number, the database will not start.

If the database is expanded with PROSTRCT ADDONLINE, the area numbers specified for any new areas must be lower than the maximum area number specified with `-maxAreas`, or the PROSTRCT ADDONLINE command will fail.

For more information, see *OpenEdge Data Management: Database Administration*.

[16] The minimum value is the current maximum area number in use for the database.

Maximum Dynamic Server (-maxport)

Use Maximum Dynamic Server (-`maxport`) to specify the highest port number in a specified range of port numbers accessible to a client.

Operating system and syntax	UNIX / Windows	**`-maxport`** ***`n`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	System dependent	System dependent	—	2000

`n`

The port number that is the highest in a specified range.

You specify the lowest port number with the Minimum Dynamic Server (-minport) on page 149 parameter. The range of port numbers defined by the Maximum Dynamic Server (-maxport) and Minimum Dynamic Server (-minport) on page 149 parameters provides client access to an OpenEdge server that is behind a firewall. Some operating systems choose transient client ports in the 32,768-to-65,535 range. Choosing a port in this range might produce unwanted results.

Maximum JTA Transactions (-maxxids)

Use Maximum JTA Transactions (-`maxxids`) to specify the number of simultaneous JTA transactions allowed.

Operating system and syntax	UNIX / Windows	**`-maxxids`** ***`n`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	32,000	100	—	100

`n`

The maximum number of simultaneous JTA transactions allowed.

JTA transactions rely on an external transaction manager to control transaction commit and rollback. OpenEdge SQL supports the Java Transaction API (JTA), enabling the database to function as a resource manager as defined in the J2EE framework architecture.

For more information on distributed transactions with JTA, see *OpenEdge Data Management: Database Administration* and *OpenEdge Data Management: SQL Reference*.

Network Message Compression (-mc)

Use Network Message Compression (-mc) to compress messages between an OpenEdge client and the OpenEdge AppServer.

Operating system and syntax	UNIX / Windows	`-mc`		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

Use Network Message Compression (`-mc`) only between OpenEdge clients and compatible servers. If the client and server are not compatible, a communication error occurs. For more information on specific compatibilities, contact Progress Software Corporation Technical Support.

When a compression-enabled client connects to an AppServer that is not compression-capable, compression is disabled for the network connection. When a compression-capable AppServer receives a compressed message from a compression-enabled client, it can decompress messages from the client and will respond to the client with a message in a compressed form.

Message compression can improve message throughput by reducing network overhead within the lower communication layers. This is more common on slower topologies, such as dial-up and WAN configurations. Progress Software Corporation recommends that you proceed with caution on faster, more optimized networks where network performance is higher. The high per-packet processing cost of enabling compression can increase network latency as it improves throughput. The net result could degrade the overall performance of the application.

Memory consistency check (-MemCheck)

Use Memory consistency check (-MemCheck) to enable memory overwrite checking for the buffer pool.

Operating system and syntax	UNIX / Windows	`-MemCheck`		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Connection, Database Server	—	—	—	—

Memory consistency check detects illegal operations in memory movement, such as a buffer copy with a negative length of data, or a memory move to an incorrect location. The purpose Memory consistency check is to prevent memory corruption.

Index and record block operations are checked prior to before-image note writing and the subsequent database block write. The length of the intended operation is checked to prevent operations that exceed database the block size.

Index block splits and index insert operations are checked at strategic points to monitor consistency.

Memory consistency checking can be enabled or disabled while your database is online with PROMON. See *OpenEdge Data Management: Database Administration* for more information.

Memory consistency check can be enabled for a single user client or for any offline utility. If the utility is read-only, such as DBANALYS, the `-MemCheck` parameter is ignored. Online utilities determine whether or not to perform memory consistency checking during execution based on the use of `-MemCheck` at broker startup or by the enablement/disablement of the memory consistency checking in PROMON.

Delayed BI File Write (-Mf)

Use Delayed BI File Write (`-Mf`) to improve performance on a heavily loaded system.

Operating system and syntax	**UNIX / Windows**	**-Mf *n***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	32,768	0	0[17]	3

n

> Any positive value delays OpenEdge from synchronously writing out to disk the last before-image (BI) file records at the end of each transaction. On UNIX systems using shared memory, it also specifies the interval that the broker process wakes up to make sure all BI file changes have been written to disk. The default is 3 for single-user batch jobs and for multi-user databases using shared memory. Otherwise, the default is 0.

Using the Delayed BI File Write (`-Mf`) parameter does not reduce database integrity. However, if there is a system failure, it is possible the last few completed transactions will be lost (never actually written to the BI file).

When running with full integrity, at the end of each transaction OpenEdge does a synchronous write to disk of the last BI file block. This write guarantees that the completed transaction is recorded permanently in the database. If the user is notified that the transaction has completed and the system or database manager crashes shortly afterwards, the transaction is not lost.

Do not set the `-Mf` parameter on a lightly loaded system with little database update activity. Under these conditions, the extra BI write is very important and does not impact performance. On a heavily loaded system, however, the BI write is less important (the BI block will be written to disk very soon anyway), and has a significant performance penalty. Setting the `-Mf` parameter to delay this extra BI write saves one write operation per transaction, which can significantly improve performance. The extra BI file write is delayed by default for batch jobs.

[17] Default is 3 for batch jobs.

If the -Mf parameter is set to a positive value, the last BI file record is only guaranteed to be written out to disk when a user logs out, or when the server or broker process terminates normally. On multi-user systems, the *n* argument determines the maximum length of time in seconds during which completed transactions can be lost.

Minimum Clients per Server (-Mi)

Use Minimum Clients per Server (-Mi) to specify the number of remote users on a server before the broker starts another server (up to the maximum number of servers).

Operating system and syntax	**UNIX / Windows**	**-Mi *n***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	—	1	—	1

n

The number of remote users on a server before the broker starts another server. See also the Maximum Servers (-Mn) on page 151 startup parameter.

This parameter pertains only to multi-user versions of OpenEdge that use shared memory. In addition, -Mi and -Mn apply only to databases accessed from remote network nodes.

As remote users enter the database, the broker process starts just one server process for each *n* remote users, until the maximum number of servers (specified by the Maximum Servers (-Mn) on page 151 parameter) is started. If you specify a value of 1, the broker starts a new server for each of the first -Mn remote users. Subsequent remote users are distributed evenly among the servers until the maximum number of users (-n) or maximum clients per server (-Ma) limits are reached.

Typically, you can leave -Mi and -Mn at their default values. If you significantly increase -Mn, you should also increase -Mi. For example, if you set -Mn to 10 to accommodate up to 40 or more remote users, increase -Mi to 3 or 4 to prevent a situation where 10 servers were started for just 10 remote users.

Minimum Dynamic Server (-minport)

Use Minimum Dynamic Server (`-minport`) to specify the lowest port number in a specified range of port numbers accessible to a client.

Operating system and syntax	**UNIX / Windows**	**`-minport n`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	—	1,025	—	1,025

n

The port number that is the lower in a specified range.

You specify the higher port number with the `-maxport` parameter. Ports below 1025 are usually reserved for system TCP and UDP. The range of port numbers defined by the `-maxport` and `-minport` parameters provides client access to an OpenEdge server that is behind a firewall. This communication is possible only when the access to the server can be limited.

Message Buffer Size (-Mm)

Use Message Buffer Size (`-Mm`) to specify the standard message buffer size, in bytes. This parameter is relevant only for network client/server connections. Message Buffer Size (`-Mm`) can be specified by both the client and the server, however the size of the message buffer specified by the server takes precedence; a value specified by the client is used as a suggestion for initial buffer allocation, but the client then adopts the server value when the connection is initiated.

Operating system and syntax	**UNIX / Windows**	**`-Mm n`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Connection, Database Server	32,600	350	—	1,024

n

The message buffer size.

OpenEdge uses message buffers to move records (messages) between servers and remote clients. Records (plus 40-byte headers) larger than the message buffer size are fragmented into multiple messages. If your database records are large, increase this parameter to avoid record fragmentation. However, if the network works more efficiently with small messages, reduce -Mm and fragment larger records.

Note: Prior to OpenEdge Release 11.6, the value of -Mm had to agree between the client and server or the connection would be refused. Going forward, in addition to not requiring agreement between client and server, clients can also connect to multiple databases with different -Mm values obtained from the server. To enforce the earlier behavior, use Enforce Mm (-enforceMm).

Maximum Memory (-mmax)

Use Maximum Memory (-mmax) to change the initial amount of memory allocated for r-code segments, in kilobytes.

Operating system and syntax	UNIX / Windows	**-mmax *n***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	65,534	1	3,096	3,096

n

The amount of memory allocated for r-code segments.

The AVM dynamically allocates space for r-code segments in the execution buffer as needed. When memory allocation reaches the value specified by the Maximum Memory (-mmax) value, the AVM writes nonactive segments to the sort file to make room for new active procedures. (the AVM writes library-stored r-code to the sort file only if you specified the PROLIB Swap (-pls) startup parameter.) If you have large procedures or deeply nested procedure calls, you can use -mmax to increase the initial size of the execution buffer to reduce disk I/O activity required to swap segments to the sort file.

The -mmax value does not limit the amount of memory available for r-code segments. If the AVM requires more space in the execution buffer and cannot regain enough space by swapping inactive segments to the sort file, the AVM allocates more memory.

The -mmax limit is a soft limit; so if your application needs to exceed the limit, the AVM automatically increases it and issues a WARNING message. The WARNING is written to the current output (if there is current output) and to the log file (LG).

You can force OpenEdge to adhere to the specified -mmax limit by starting the session with the Hardlimit (-hardlimit) on page 119 startup parameter. When you use the -hardlimit startup parameter, OpenEdge issues the WARNING message when you exceed the Maximum Memory limit. It also issues a message that a resource limit was reached and raises an untrappable STOP condition. Note also that -hardlimit also enforces the limits set by the Directory Size (-D) on page 100, the Local Buffer Size (-l) on page 129, and the Nested Blocks (-nb) on page 157 startup parameters. The STOP condition is raised when any of the specified limits is reached.

Use Statistics (-y) on page 224 and Segment Statistics (-yd) on page 226 to see segment allocation information.

Maximum Servers (-Mn)

Use Maximum Servers (`-Mn`) to limit the number of remote user servers that can be started by the broker process.

Note: The maximum value for the number of servers that may be started for a database is limited by available resources of the operating system.

Operating system and syntax	UNIX / Windows	**`-Mn n`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	—[18]	1	—	4

n

> The maximum number of remote client servers that can be started on the system. The value specified is always incremented by 1 to provide a server for the primary login broker.

The performance trade-off to consider is swapping overhead for many servers versus overloading (slowing down) a server with too many clients. For more information on controlling memory use, see *OpenEdge Data Management: Database Administration*.

Also, use Minimum Clients per Server (-Mi) on page 148 to adjust the actual number of servers in use. See the Maximum Clients Per Server (-Ma) on page 143 and Minimum Clients per Server (-Mi) on page 148 startup parameters for more information.

[18] Limited by available resources.

Servers Per Protocol (-Mp)

Use Servers per Protocol (`-Mp`) with Secondary Login Broker (-m3) on page 143 in database networks that use more than one network protocol.

Operating system and syntax	**UNIX / Windows**	**`-Mp n`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	value of -Mn	1	—	value of -Mn

n

The number of servers a broker can start.

This parameter limits the number of servers that the broker can start to serve remote users for any one protocol. The total number of servers for all protocols is still limited by the Maximum Servers (-Mn) on page 151 parameter.

Maximum Servers Per Broker (-Mpb)

Use Maximum Server Per Broker (`-Mpb`) to specify the maximum number of servers that multiple brokers can start to serve remote users for any one protocol.

Operating system and syntax	**UNIX / Windows**	**`-Mpb n`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	—	—	—	—

n

The number of servers multiple brokers can start.

For more information, see *OpenEdge Data Management: Database Administration*.

Record Buffer Size (-Mr)

Use Record Buffer Size (`-Mr`) to improve performance by varying the standard record buffer size.

Operating system and syntax	UNIX / Windows	**-Mr *n***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	32,000	96	—	1,012

n

> The standard record buffer size in bytes. The default value of 1,012 bytes is appropriate in nearly all cases.

On some systems, you can use Record Buffer Size (`-Mr`) to improve performance by varying the standard record buffer size. However, the default value of 1,012 bytes is appropriate in nearly all cases.

This parameter determines the maximum size of a record buffer the ABL client can hold in its buffer cache. If you need to fetch a row that is larger than this value, then the AVM allocates a new record buffer to hold the larger record and frees the buffer after the record buffer goes out of scope. If you set this parameter too low, the client will spend more time allocating and freeing memory. Set this parameter to be larger than the average size record the application will read from the database.

User MUX Latches (-mux)

Use User MUX Latches (`-mux`) to control the granularity of access to large database resources in shared memory.

Operating system and syntax	UNIX / Windows	**-mux *n***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server, Client Connection	1	0	—	1

Do not use this parameter unless directed to do so by Progress Software Technical Support.

Shared-memory Overflow Size (-Mxs)

Use Shared-memory Overflow (`–Mxs`) to replace the default value of the shared-memory overflow area; it does not increase it.

Note: Starting in Release 10.1C, the OpenEdge RDBMS expands the range of internal validations used to ensure database consistency in both index and data blocks during forward processing. Validations using PROUTIL can be run online as part of routine health checks. For more information, see *OpenEdge Data Management: Database Administration*.

Operating system and syntax	**UNIX / Windows**	**–Mxs *n***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	Varies[19]	1	—	Varies

n

The size of the shared-memory overflow area in kilobytes.

The overflow area is appended to the shared-memory area. This parameter is relevant only on shared-memory systems.

If the overflow area is too small, OpenEdge exits with the following message:

```
SYSTEM ERROR: Out of free shared memory. Use -Mxs to increase.
```

Depending on the operating system, OpenEdge rounds the shared-memory area size to the next 512-byte or 4K boundary.

[19] The maximum is limited only by the size of the signed integer data type on the system. If your system uses a 4 byte integer, then the formula is: 16KB + (n*300). For all other systems, the formula is: 16KB + (n*400).

Number of Users (-n)

Use Number of Users (`-n`) to limit the total number of users below the level that overloads the database server on systems that do not use shared memory.

Operating system and syntax	UNIX / Windows	**-n *n***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	10,000	—	—	20

n

The maximum number of OpenEdge users on the system. After *n* users have connected to the OpenEdge database, additional user startup attempts are rejected.

Note: The actual number of allowed users might be smaller than the value of this parameter. The actual number of allowed users must be equal or less than the result of the Maximum Servers (-`Mn`) multiplied by the Maximum Clients Per Server (-`Ma`), using whole number values for those parameters.

On systems that do not use shared memory, use Number of Users (`-n`) to limit the total number of users below the level that overloads the database server.

On shared-memory systems, `-n` must be high enough to include local and remote users as well as background writers (APWs, BIWs, and AIWs), PROWDOG processes, and PROMON sessions. For more information, see the Minimum Clients per Server (-Mi) on page 148 and Maximum Servers (-Mn) on page 151 startup parameters.

Nap Time Increment (-napinc)

Use Nap Time Increment (`-napinc`) to specify the nap time increment for Enterprise database licenses only.

Operating system and syntax	UNIX / Windows	**-napinc *n***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	—	—	—	10

Do not use this parameter unless directed to do so by Progress Software Technical Support.

Nap Maximum (-napmax)

Use Nap Maximum (`-napmax`) to specify the maximum time (in milliseconds) to sleep (nap) after the -spin value is exhausted due to failure to acquire a latch.

Operating system and syntax	**UNIX / Windows**	**`-napmax n`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	—	—	—	250

n

The maximum time (in milliseconds) to sleep (nap).

Increasing the value of `-napmax` can decrease CPU utilization, and can decrease individual user performance while improving performance of the overall system.

Testing by Progress Software has determined that 250 milliseconds is a good default. Changing the value of `-napmax` away from the default is not recommended.

Nap Time Steps (-napstep)

Use Nap Time Steps (`-napstep`) to specify the steps between the nap increments for Enterprise database licenses only.

Operating system and syntax	**UNIX / Windows**	**`-napstep n`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	—	—	—	5

Do not use this parameter unless directed to do so by Progress Software Technical Support.

Nested Blocks (-nb)

Use Nested Blocks (`-nb`) to limit the maximum number of nested procedure blocks allowed.

Operating system and syntax	UNIX / Windows	**`-nb n`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	20000	20	100	100

`n`

The maximum number of nested blocks.

By default, the maximum number of nested blocks is 100. Nested block entries are allocated in memory; each requires 32 bytes. Therefore, decrease `-nb` only if memory is severely limited.

The `-nb` limit is a soft limit. If your application needs to exceed the limit, the AVM issues a `WARNING` message and automatically increases the number of nested procedure blocks allowed. The `WARNING` is written to the current output (if there is current output) and to the log file (`LG`).

You can force the AVM to adhere to the specified `-nb` limit by starting the session with the Hardlimit (-hardlimit) on page 119 startup parameter. When you use the `-hardlimit` startup parameter, the AVM issues the `WARNING` message when you exceed the Nested Blocks (`-nb`) limit. It also issues a message that a resource limit was reached and raises an untrappable `STOP` condition. Note that `-hardlimit` also enforces the limits set by the Directory Size (-D) on page 100, the Local Buffer Size (-l) on page 129, and the Maximum Memory (-mmax) on page 150 startup parameters, so the `STOP` condition is raised when any of the specified limits is reached.

Login Governor (-nGovernor)

Use Login Governor (`-nGovernor`) to specify the maximum number of users who can login for a tenant in a multi-tenant database. The valid value for the Login Governor ranges from 1 to 32000.

Operating system and syntax	UNIX / Windows	**`-nGovernor n`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	32000	1	—	—[20]

[20] The default is the value of the `-n` startup parameter, implying there is no per-tenant limit.

n

The maximum number of users.

Note: Login governor applies to the default tenant and a regular tenant. This governor does not apply to super-tenants and non multi-tenant databases.

Login governor prevents a tenant from using all the database connections, and prevents other tenants from logging in. When a user attempts to log into a database, and the available login resource allocation for the tenant has been exceeded, an error message is recorded in the database log as:

```
Attempt to exceed tenant login governor for tenant <tenant ID>. Increase -n
or -nGovernor.
```

The Startup file Virtual System Table (VST) contains a field for the Login Governor startup parameter. For more information on Startup file, see *OpenEdge Data Management: Database Administration*.

The PROMON utility helps you to monitor and change Login Governor values. For more information on how the PROMON utility monitors online Login Governor, see *OpenEdge Data Management: Database Administration*.

No Lock (-NL)

Use No Lock (-NL) to have all record retrieval statements default to NO-LOCK record access.

Operating system and syntax	**UNIX / Windows**	**-NL**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

In order to get the NO-LOCK default for any procedure, it must be compiled in an OpenEdge session started with -NL. Running OpenEdge with -NL has no affect on precompiled procedures.

When a transaction ends, EXCLUSIVE locks are downgraded to SHARE-LOCK, regardless of the -NL parameter.

Message Wait (-Nmsgwait)

Use Message Wait (`-Nmsgwait`) to indicate the number of seconds a server waits for a remote network message before checking for other events such as a database shutdown or forced disconnect of a remote client.

Operating system and syntax	**UNIX / Windows**	**`-Nmsgwait` *`n`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	255	1	—	2[21]

`n`

The number of seconds a server waits for a remote network message.

No Auto Result-list (-noautoreslist)

Use No Auto Result-list (`-noautoreslist`) to avoid building result-lists for static non-scrolling queries.

Operating system and syntax	**UNIX / Windows**	**`-noautoreslist`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

A static non-scrolling query is one that is:

- Defined using the `DEFINE QUERY` statement without the `SCROLLING` option
- Opened using the `OPEN QUERY` statement (but not previously defined using the `DEFINE QUERY` statement)
- Not associated with a browse widget (and has not been changed into a scrolling query by the `DEFINE BROWSE` statement)

When you specify this startup parameter, you cannot use the `GET PREV`, `GET LAST`, `REPOSITION`, or `BROWSE` methods or statements with static non-scrolling queries. If you do, the AVM generates an error. You can use the `GET-FIRST( )` method and `GET FIRST` statement only on newly opened queries, and you can use the `GET NEXT` statement and `GET-NEXT( )` method freely.

[21] The default of 2 produces pre-Release 11.1 behavior.

If you specify this startup parameter and you open a static non-scrolling query with preselect or sort, the AVM still builds a result-list in order to resolve the query.

Specifying this startup parameter has no affect on dynamic queries, since they are scrolling queries by default.

Specifying this startup parameter can improve the performance of operations on queries.

No Domain Support With the CAN-DO Function (-nocandodomain)

The No Domain Support With the `CAN-DO` Function (`-nocandodomain`) parameter forces `CAN-DO` to treat "`@`" as a regular character instead of as a delimiter for the domain in a fully qualified user ID.

Operating system and syntax	**UNIX / Windows**	**`-nocandodomain`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

By default, the `CAN-DO` function treats the "`@`" symbol as the domain delimiter in a fully qualified user ID. You can use the `-nocandodomain` parameter to force `CAN-DO` to treat the "`@`" symbol as a regular character. For example:

- When `-nocandodomain` is not in use, the statement `CAN-DO("abc","abc@")` evaluates to `TRUE` because both strings are interpreted as user `abc` in the blank domain.
- When `-nocandodomain` is in use, the statement `CAN-DO("abc","abc@")` evaluates to `FALSE`.

You can achieve the same effect within the session by setting the `CAN-DO-DOMAIN-SUPPORT` attribute on the `SECURITY-POLICY` handle to `FALSE`. Similarly, setting the `CAN-DO-DOMAIN-SUPPORT` attribute to `TRUE` will override `-nocandodomain`. See *OpenEdge Development: ABL Reference* for more information.

No Check Temp-table Names (-nochkttnames)

Use the No Check Temp-Table Names (`-nochkttnames`) parameter to suppress the limited checking of temp-table column names when passing temp-table as parameters to procedures.

Operating system and syntax	**UNIX / Windows**	**`-nochkttnames`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

When passing temp-table as a parameter to a procedure, the calling procedure's temp-table and the called procedure's temp-table must match with respect to the number of columns, the data type of each column, and the number of extents of each column (for columns with extents).

The AVM provides limited checking of temp-table column names when passing temp-tables as parameters to procedures. That is, for any given column:

- If the name of a column in the calling procedure's temp-table does not match any column defined in the called procedure's temp-table, the AVM processes the column.
- If the name and position of a column in the calling procedure's temp-table matches a column defined in the called procedure's temp-table, the AVM processes the column.
- If the name of a column in the calling procedure's temp-table matches a column defined in the called procedure's temp-table, but the positions do not match, the AVM reports an error and the called procedure does not run.

Use the No Check Temp-table Names (`-nochkttnames`) parameter when you want the AVM to process the temp-table without verifying column names.

Note: Temp-table column names are case-insensitive.

No Colons for Side Labels (-nocolon)

Use No Colons for Side Labels (`-nocolon`) to prevent colons from being automatically appended to side labels for static widgets.

Operating system and syntax	**UNIX / Windows**	**`-nocolon`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

At compile time, colons are automatically appended to side labels for static widgets. To create side labels without colons, compile the r-code with -nocolon in effect. Running OpenEdge with -nocolon does not affect precompiled procedures. Side labels created by code that was not compiled with -nocolon in effect will include colons even if the current session is running with -nocolon active.

No Evaluation of Properties (-noevalprops)

Use the No Evaluation of Properties (-noevalprops) parameter to prevent the OpenEdge Debugger from evaluating ABL properties during a debugging session.

Operating system and syntax	UNIX / Windows	**-noevalprops**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Connection	—	—	—	—

The reason for using this parameter is to avoid errors that can occur in the Debugger when it executes the property's accessor (GET) method. Such errors might occur, for example, if the Debugger attempts to evaluate the property before the object is fully constructed. Certain side effects of the property's GET method can also lead to Debugger errors.

If you use this startup parameter, the Debugger displays "** Property evaluation disabled **" for any property that has a GET method with an implementation. The parameter has no effect on ABL properties with empty GET methods, which are always evaluated.

If property evaluation is disabled, you can use the **Dataview** dialog box to inspect the value of a property. For more information, see *OpenEdge Development: Debugging and Troubleshooting*.

No Garbage Collection (-nogc)

Use No Garbage Collection (-nogc) to disable automatic garbage collection of class-based objects. By default, the AVM automatically deletes .NET objects and ABL objects when there are no remaining references to the object. When -nogc is specified, the application assumes the responsibility of deleting .NET objects and ABL objects explicitly, using the DELETE OBJECT statement. If -nogc is specified, .NET objects and ABL objects that are not explicitly deleted are deleted when the session ends, and the AVM does not invoke any destructors for the class.

Operating system and syntax	Windows	**-nogc**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

For more information on garbage collection, see *OpenEdge Development: ABL Reference* and *OpenEdge Development: Object-oriented Programming*.

No Host Verify (-nohostverify)

Use No Host Verify (–`nohostverify`) to turn off host verification for a Secure Sockets Layer (SSL) connection to a database server.

Operating system and syntax	**UNIX / Windows**	**`-nohostverify`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Connection	—	—	—	—

Without this parameter specified, the client compares the host name specified in the connection with the Common Name specified in the server certificate, and raises an error if they do not match. With this parameter specified, the client never raises the error.

For more information, see *OpenEdge Getting Started: Core Business Services - Security and Auditing*.

No Inactive Indexes (-noinactiveidx)

Use No Inactive Indexes (-`noinactiveidx`) to exclude inactive indexes from the compilation of `WHERE` clauses, as well as `LIKE` clauses in `DEFINE TEMP-TABLE` statements.

Operating system and syntax	**UNIX / Windows**	**`-noinactiveidx`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

No Auto-Increment Warnings (-noincrwarn)

Use No Auto-Increment Warnings (`-noincrwarn`) to suppress messages (`5407`) through (`5410`).

Operating system and syntax	UNIX / Windows	**-noincrwarn**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

The auto-increment messages, messages (`5407`) through (`5410`), warn about the automatic resource usage increases by the Maximum Memory (-mmax) on page 150, Nested Blocks (-nb) on page 157, Local Buffer Size (-l) on page 129, and Directory Size (-D) on page 100 startup parameters.

Note: Using the `SESSION:SUPPRESS-WARNINGS` attribute will suppress all warning messages during a session, including auto-increment warnings. By using the No Auto-Increment Warnings (-noincrwarn) startup parameter without the `SESSION:SUPPRESS-WARNINGS` attribute, all normal warning messages display except the auto-increment warning messages.

Caution: The auto-increment warnings may indicate a memory leak in your application, resulting in excessive resource usage. The No Auto-Increment Warnings (-noincrwarn) startup parameter does not stop this excess resource usage; it stops the reporting of this condition.

Index Hint (-noindexhint)

Use Index Hint (`-noindexhint`) to stop a DataServer for ORACLE from providing index hints to the ORACLE DBMS, which the DataServer does by default.

Operating system and syntax	UNIX / Windows	**-noindexhint**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
DataServer	—	—	—	—

Generally, index hints improve performance, but ORACLE's responses to hints varies among releases. This parameter is valid only when using the DataServer for ORACLE.

No INT64 (-noint64)

Use No INT64 (-noint64) to change the data type of long integer constants to be DECIMAL as opposed to INT64. Does not affect references to variables, fields, or expressions if they are declared as INT64.

Operating system and syntax	UNIX / Windows	-noint64		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

If you specify the -noint64 startup parameter:

- The functions that are compiled as INT64 (CURRENT-VALUE, DYNAMIC-CURRENT-VALUE, DYNAMIC-NEXT-VALUE, ETIME, GET-POINTER-VALUE, GET-SIZE, INTERVAL, NEXT-VALUE, and SEEK) are compiled as INTEGER.
- INT64 constants are compiled as DECIMAL.
- ABL source code references to INT64 variables and fields will not generate errors.
- Code that subtracts two DATETIMEs generates a compile-time error.
- The XML schema long data type (64-bit signed integer) maps to the ABL DECIMAL data type rather than the INT64 data type. This mapping applies to:
  - The bprowsdldoc and bproxsdto4gl command line utilities
  - The Web Services Client runtime product

Note: Using the -noint64 startup parameter does not does not affect performance or memory usage since the space and time taken up by int64 calculations is the same as for integer calculations.

Server Join (-nojoinbysqldb)

Use Server Join (-nojoinbysqldb) to specify that the client evaluate and performs queries that have joins.

Operating system and syntax	UNIX / Windows	-nojoinbysqldb		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
DataServer	—	—	—	—

This might slow performance, but provides results that are consistent with OpenEdge behavior. It also allows you to run DataServer applications on Version 9 clients with Version 8 servers. It overrides the DataServer default behavior, which is to instruct the non-OpenEdge server to perform the join.

Note: Use `-nojoinbysqldb` at compile time. It has no effect at run time.

No routine in WHERE Clause (-noroutineinwhere)

Use No ROUTINE in WHERE (`-noroutineinwhere`) to restrict the `WHERE` clause from containing a class method, a user-defined function, or a class property which implements a `GET` method using ABL statements. If they are found in a `WHERE` clause, the compiler raises an error.

Operating system and syntax	**UNIX / Windows**	**`-noroutineinwhere`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

No Session Cache (-nosessioncache)

Use No Session Cache (`-nosessioncache`) to disable SSL session caching.

Operating system and syntax	**UNIX / Windows**	**`-nosessioncache`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	—	—	—	—

Session caching allows a client to reuse a previously established session if it reconnects prior to the session cache timeout expiring. Use No Session Cache (`-nosessioncache`) to disable this feature. Session caching is enabled by default.

No Session Reuse (-nosessionreuse)

Use No Session Reuse (`-nosessionreuse`) when you do not want to reuse a Secure Sockets Layer (SSL) session ID to reconnect to the same SSL-enabled database server.

Operating system and syntax	**UNIX / Windows**	**`-nosessionreuse`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Connection	—	—	—	—

For more information, see *OpenEdge Getting Started: Core Business Services - Security and Auditing*.

Note: OpenEdge SSL turns on SSL session reuse by default. So, after the initial connection to a given host (`-H`) and port (`-S`), each subsequent connection to the same host and port restarts the SSL session and ignores any different connection parameters that are specified for the subsequent connection, including `-nosessionreuse`. If you want to change SSL socket options (such as `-nohostverify`) for each subsequent connection to a given host and port, be sure to specify the `-nosessionreuse` parameter on the initial SSL socket connection to that same host and port.

SELECT Pass Through Disable (-noSQLbyserver)

Use SELECT Pass Through Disable (`-noSQLbyserver`) to turn off SELECT Pass Through for the current OpenEdge session.

Operating system and syntax	**UNIX / Windows**	**`-noSQLbyserver`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
DataServer	—	—	—	—

Use with DataServers for ODBC and ORACLE.

No UDF in WHERE Clause (-noudfinwhere)

Use No UDF in WHERE Clause (`-noudfinwhere`) to restrict the invocation of user-defined functions and methods from within `WHERE` clauses. When specified, invoking a user-defined function or method from within a `WHERE` clause generates a compiler error.

Operating system and syntax	**UNIX / Windows**	**`-noudfinwhere`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

Fractional Separator (-numdec)

Use Fractional Separator (`-numdec`) to change the character that represents a number's decimal point in formatted text. The default decimal point is the period (`.`).

Operating system and syntax	**UNIX / Windows**	**`-numdec`** ***`numeric-value`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	. (period)	. (period)

`numeric-value`

The numeric value of the character that represents the decimal point.

The AVM converts *`numeric-value`* to its character equivalent using the code page corresponding to the Internal Code Page (-cpinternal) on page 91 startup parameter.

The decimal point cannot be represented by any of the following:

- The characters `B C D R Z z 0 1 2 3 4 5 6 7 8 9 + - < > ( ) * ?`
- The space character
- Any multi-byte character

In the following example, `-numdec` is used to change the decimal point in formatted text to a comma (which, in the ISO8859-1 code page, has a numeric value of 44):

```
-numdec 44
```

Number of Log Files to Keep (-numlogfiles)

Use Number of Log Files to Keep (`-numlogfiles`) to specify the total number of rolled over log files to keep on disk at any one time, across OpenEdge sessions, including the current log file.

Operating system and syntax	**UNIX / Windows**	**`-numlogfiles`** ***`n`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	3	3

`n`

The number of log files to keep. Valid values are:

- **0** — This means there is no limit on the number of log files to keep.
- **2 or greater** — The default is 3.

  If you specify a value of 1, a run-time error occurs and the OpenEdge process terminates.

Use the Log Threshold (-logthreshold) on page 140 startup parameter to specify the file size at which OpenEdge renames and saves log files.

Use the Client Logging (-clientlog) on page 86 startup parameter to specify a log file name for ABL clients.

Use the DataServer Logging (-dslog) on page 108 startup parameter to specify a log file name for server context of a DataServer client or agent.

You also can use the `NUM-LOG-FILES` attribute on the `LOG-MANAGER` system handle to specify the number of log files to keep. For more information, see *OpenEdge Development: ABL Reference*.

Thousands Separator (-numsep)

Use Thousands Separator (`-numsep`) to change the character that represents a number's thousands separator in formatted text. The default thousands separator is the comma (`,`).

Operating system and syntax	**UNIX / Windows**	**`-numsep`** ***`numeric-value`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	, (comma)	, (comma)

numeric-value

The numeric value of the character that represents the thousands separator.

The AVM converts *numeric-value* to its character equivalent using the code page corresponding to the Internal Code Page (-cpinternal) on page 91 startup parameter.

The thousands separator cannot be represented by any of the following:

- The characters `B C D R Z z 0 1 2 3 4 5 6 7 8 9 + - < > ( ) * ?`
- Any multi-byte character

Note: You can represent the thousands separator with the space character.

In the following example, `-numsep` is used to change the thousands separator in formatted text to an apostrophe (which, in the ISO8859-1 code page, has a numeric value of 39):

```
-numsep 39
```

Printer (-o)

Specifies the printer to use.

Operating system and syntax	**UNIX / Windows**	**-o *printername***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	System dependent[22]	System dependent[22]

printername

Identifies the printer to use when processing the `OUTPUT TO PRINTER` statement in procedures.

In Windows, use Printer (-o) followed by the name of the printer port. For example:

```
-o LPT1
```

If you do not use the `-o` parameter, the AVM uses the printer specified as the default printer.

[22] See defaults listed below.

On UNIX, use Printer (-o) followed by the name of the spooler and any necessary parameters. If you do not use the `-o` parameter, the AVM defaults to `lp` (System V) or `lpr` (BSD). If you are passing arguments with this parameter and are using a script such as PRO or one of your own, enclose the entire string in quotes so the argument to `-o` is returned as a single value when passed to the OpenEdge module. For example:

```
-o "lp -s"
```

The AVM uses the spooler or printer port named at execution time rather than using the name at compile time. That way, precompiled procedures work regardless of which spooler or printer port is in effect.

To use a print spooler with spooler parameters, use `OUTPUT TO PRINTER` and specify the spooler parameters with `-o`, as in the following commands:

```
OUTPUT THROUGH lp spooler-parameters PAGED.
```

```
OUTPUT THROUGH lpr spooler-parameters PAGED.
```

Outer-join Mode (-ojmode)

Use the Outer-join Mode (`-ojmode`) parameter to specify the mode in which mixed inner and left outer joins, in queries of three or more joined tables, are processed.

Operating system and syntax	UNIX / Windows	**-ojmode *n***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	2	1	1	1

n

An `INTEGER` specifying the join mode.

Mixing inner and left outer joins lets you filter and reduce the amount of data you see on the left side of your left outer joins. When mixing these two types of join, inner joins take precedence over left outer joins. That is, the last inner join in a query causes the results of a prior outer join in the query to be like an inner join. This is because any rows that contain missing data from a prior left outer join are eliminated by the following inner join, if the data from that inner join is also missing. If the subsequent inner join has data, rows containing missing data from the prior left outer join are not eliminated.

If you set the value of `-ojmode` to `1`, mixed inner and left outer joins in queries of three or more joined tables are processed as described above. Use this join mode when you want to see rows that contain missing data from a prior left outer join and existent data from a subsequent inner join. This is the default join mode.

If you set the value of `-ojmode` to `2`, left outer joins take precedence over inner joins. That is, any rows that contain missing data from a prior left outer join are not eliminated by a subsequent inner join, whether the data from that inner join is missing or not. Use this join mode when you want to see rows that contain missing data from both a prior left outer join and a subsequent inner join.

Note: In any query, keeping your inner joins contiguous on the left with any left outer joins contiguous on the right will produce the same result in either join mode.

Storage Object Cache Size (-omsize)

Use Storage Object Cache Size (`-omsize`) to specify the size of the object cache for all database objects.

Operating system and syntax	**UNIX / Windows**	**`-omsize n`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	65535	0	—	1024

Password (-P)

Use Password (`-P`) together with the User ID (-U) on page 218 connection parameter to specify the user ID and user account password when connecting to an OpenEdge RDBMS, either at ABL application startup or when executing the ABL `CONNECT` statement.

Operating system and syntax	**UNIX / Windows**	**`-P password`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Connection	—	—	—	—

`password`

The password of the user account connecting to the database.

By default, *password* is a cleartext string. However, you may require an encrypted version of *password* when security is a concern (for example, when including passwords in startup (.pf) files). You can generate and use an encrypted password by doing the following:

1. Start Proenv.
2. Create the encrypted password using the genpassword utility. For example:

```
proenv>genpassword -password "clear_text" xxxxxxxxx
```

   where clear_text is a cleartext password. xxxxxxxxxx is the output of genpassword -password, and represents an encrypted version of the password.

3. Add the prefix oech1 to the encrypted password using the double colon (::) separator (oech1::xxxxxxxxxx). The prefix oech1 specifies the encryption algorithm (oec), the encoded password format (h), and the encryption keypad (1). The prefix is necessary so that OpenEdge will properly interpret the string as an encrypted value. Without the prefix, the string would be processed as if it were cleartext and authentication would fail.

For information on generating and using encrypted passwords in the ABL, see the "Password encryption" section in *OpenEdge Development: Programming Interfaces*.

OpenEdge authenticates the user ID and password specified by the -U and -P parameters against the user account system associated with the user's OpenEdge security domain. If a user account is found that matches the specified user ID and password, OpenEdge completes the database connection with the specified connection identity. Otherwise, the user is not allowed to connect to the database. If the database is multi-tenant, a successful connection also sets the user's tenancy.

For more information on authenticating users when connecting to a database, see the User ID (-U) on page 218 parameter entry.

Note:

- The -P value is the same as the value of the PRIMARY-PASSPHRASE attribute on a client-principal object handle used in an ABL application. It has nothing to do with the domain access code that you specify for the configuration of a user's OpenEdge security domain.
- With certain DataServers, the -P and -U parameters pass DataServer login information to the foreign (non-OpenEdge) database. For more information, see your DataServer documentation.
- If you are connecting to an encrypted database, you might have to provide a key store passphrase as well as a user password, depending on how the database is started. For more information on encrypted databases and how to connect to them, see the sections on Transparent Data Encryption in *OpenEdge Getting Started: Core Business Services - Security and Auditing* and *OpenEdge Data Management: Database Administration*.

Startup Procedure (-p)

Use Startup Procedure (`-p`) to specify a procedure that executes at the start of an OpenEdge session.

Operating system and syntax	**UNIX / Windows**	**`-p` *`filename`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	_edit.p	_edit.p

`filename`

The name of the procedure to run when starting OpenEdge.

If the startup procedure is located in a procedure library, the procedure must be referenced explicitly as a member of the library. For example:

```
-p "/usr/app/sharedlib.pl<<startup.r>>"
```

Note that the quotation marks are necessary to prevent parser errors as a result of the "<<" symbols.

A common use for the `-p` parameter is to run an ABL procedure that displays a main application menu. The user can then choose options from the menu to run other ABL procedures. As a built-in security measure, when users press **CTRL+C** while running a procedure specified by `-p`, the AVM prevents the users from accessing the Procedure Editor by rerunning the specified procedure (unless the application explicitly allows Procedure Editor access).

If you use the Batch (-b) on page 74 startup parameter (`MBPRO` or `BPRO`), also use the `-p` parameter.

Parameter (-param)

Use Parameter (`-param`) to specify a character string that can be accessed from ABL procedures.

Operating system and syntax	**UNIX / Windows**	**`-param` *`string`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

string

A character string that supplies information to an ABL application. Can also be a comma-separated list of files to open in Procedure Editor buffers.

You can also use this parameter to specify one or more files to load into Procedure Editor buffers. The Procedure Editor only loads the specified files when you use this parameter with a startup command that places you directly in the Procedure Editor. The files are not loaded when you use the parameter with a startup command that places you in the OpenEdge Application Development Environment (ADE) desktop. Within an ABL procedure, you can access the string by reading the `PARAMETER` attribute of the `SESSION` system handle.

Prompt for Passphrase (-Passphrase)

Use Prompt for Passphrase (`–Passphrase`) to open an encryption-enabled database that is configured for manual start or to override the autostart passphrase configuration.

Operating system and syntax	**UNIX / Windows**	**`-Passphrase`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server, Client Connection	—	—	—	—

Database utilities and servers, and non-networked self-service clients indicate that they require a prompt for key store authentication by adding Prompt for Passphrase (`–Passphrase`) to the command line. Prompt for Passphrase (`–Passphrase`) prompts for the key store passphrase. If you cannot successfully authenticate the key store, you can not open the database.

For more information on Prompt for Passphrase, and Transparent Data Encryption in general, see *OpenEdge Getting Started: Core Business Services - Security and Auditing* and *OpenEdge Data Management: Database Administration*.

Pending Connection Time (-PendConnTime)

Use Pending Connection Time (-`PendConnTime`) to prevent connection errors caused by network problems.

Operating system and syntax	**UNIX / Windows**	**`-PendConnTime`** ***`n`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	—	—	—	0

n

Time, in seconds, allowed the client to connect to a server.

When a client attempts to connect to an OpenEdge database, the RDBMS assumes that if the client can reach the broker, it can also reach the server. Network problems or incorrect configuration of network devices can prevent the client from reaching the server. In such an instance, the broker, unaware that the client's connection failed, continues to increment its count of connected users. To prevent this problem, OpenEdge brokers use a "reservation" count on each server and they increment this count whenever they redirect a client to that server.

When `-PendConnTime` is used, the servers examine the timestamp on the latest reservation and, if the Pending Connection Time period has elapsed, the broker assumes that the client has failed to connect. The broker then clears the reservation, making the server available for new clients.

Parameter File (-pf)

Use Parameter File (`-pf`) to name a parameter file that includes any number of startup parameters to run OpenEdge.

Operating system and syntax	**UNIX / Windows**	**-pf *filename***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Connection, Client Session, Database Server, DataServer	—	—	—	—

filename

The name of a parameter file that contains startup parameters to run OpenEdge.

This parameter is especially useful if you regularly use the same parameters to start OpenEdge, or if more parameters are specified than can fit on the command line. This parameter can be included within the parameter file itself to reference another parameter file.

Use multiple instances of `-pf` to name multiple parameter files. This allows you to specify application-specific parameters in one parameter file, database-specific parameters in a second parameter file, and user-specific parameters in yet another file.

See Introduction on page 21 for more information about parameter files.

Database Service Communication Area Size (-pica)

Use Database Service Communication Area Size (`-pica`) to set the size of the database service communications area in Kb.

Operating system and syntax	**UNIX / Windows**	**`-pica`** ***n***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	1000000	4	—	16

The database service communications area is used to store after-image block write notifications intended for OpenEdge Replication.

Pin Shared Memory (-pinshm)

Using the Pin Shared Memory (`-pinshm`) parameter prevents the database engine from swapping shared memory contents to disk, which can help improve performance.

Operating system and syntax	**UNIX / Windows**	**`-pinshm`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	—	—	—	—

PROLIB Memory (-plm)

Use PROLIB Memory (`-plm`) to allocate a 512byte cache for a standard library directory.

Operating system and syntax	**UNIX / Windows**	**`-plm`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

Otherwise, when using a standard library, the library's internal directory is loaded into memory. The directory is used to access the r-code files stored in the library. This parameter slows the library's speed of access but increases available memory. If you specify `-plm` with memory-mapped libraries, the AVM ignores it. For more information on r-code libraries, see *OpenEdge Deployment: Managing ABL Applications* and *OpenEdge Getting Started: ABL Essentials*.

PROLIB Swap (-pls)

Use PROLIB Swap (`-pls`) to direct the AVM to store r-code segments from a standard procedure library locally in the r-code swap file, rather than the usual behavior.

Operating system and syntax	**UNIX / Windows**	**`-pls`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

Usually, OpenEdge opens the library file and loads the required r-code segments into the r-code execution buffer. When OpenEdge needs to make space, the segment is removed from the execution buffer. Later, if that segment is needed again, OpenEdge re-reads the segment from the open library which remains in the execution buffer until the end of your OpenEdge session or until you remove the library from the PROPATH.

This process is generally faster than loading the r-code segment from the r-code swap file. However, if the procedure library is accessed over a network, reading r-code segments once and storing them locally in the r-code swap file might be more efficient.

Note: Codepage-converted text segments from a standard procedure library are always swapped to the r-code swap file, whether or not you specify `-pls`.

If you specify `-pls` with memory-mapped procedure libraries, the AVM ignores it. For more information on procedure libraries, see *OpenEdge Deployment: Managing ABL Applications* and *OpenEdge Getting Started: ABL Essentials*.

Fast Schema Change (-populate)

Use Fast Schema Change (`-populate`) to turn off fast schema change when you are adding fields to a table. Fast schema change is the default behavior.

Operating system and syntax	**UNIX / Windows**	**`-populate`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Connection	—	—	—	—

In earlier versions of OpenEdge, adding a field to an existing database table is a time-consuming process for large databases. The DBMS updated the data in each row of the table to reflect the newly added field. Now OpenEdge provides a fast schema change when you are adding fields to a table when all qualifying conditions are met. If all conditions are not met, adding a field causes OpenEdge to update every row in the table.

Prefetch delay (-prefetchDelay)

Use Prefetch delay (`-prefetchDelay`) to enable a delay when sending the first network message for queries with prefetch capabilities.

Operating system and syntax	**UNIX / Windows**	**`-prefetchDelay`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	—	—	—	—

When you specify Prefetch delay, the Prefetch Num Recs (-prefetchNumRecs) on page 180 and Prefetch factor (-prefetchFactor) on page 180 parameters determine the duration of the delay. Prefetch delay is off by default, indicating the that first network message is sent as soon as it contains the first record of the query.

Prefetch factor (-prefetchFactor)

Use Prefetch Factor (-`preFetchFactor`) to establish a percentage of a network message required to contain prefetched data before sending the message to a waiting remote client.

Operating system and syntax	UNIX / Windows	**`-prefetchFactor`** ***`n`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	100	0	—	0[23]

`n`

The percent of a message buffer to fill before sending it to a waiting remote client.

The number of records in a network buffer is also influenced by the Prefetch Num Recs (-prefetchNumRecs) on page 180 parameter. See the *OpenEdge Data Management: Database Administration* for a discussion of the interaction between the parameters.

Prefetch Num Recs (-prefetchNumRecs)

Use Prefetch Num Recs (-`prefetchNumRecs`) to increase the number of records placed in a network message before it is sent to a remote client.

Operating system and syntax	UNIX / Windows	**`-prefetchNumRecs`** ***`n`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	32766	1	—	16[24]

`n`

The number of prefetch records to put in a network message before sending the message to a waiting remote client.

Setting Prefetch Num Recs (-`prefetchNumRecs`) may cause a waiting remote client to wait longer for full network messages containing prefetched records, but overall system performance is expected to improve by saving on network traffic. Another possible side-effect is that an individual remote client see might see "choppy" responses from the server.

[23] The default of 0 give pre-Release 11.1 behavior

[24] Prior to Release 11.1, 16 was the static setting.

The number of records in a network buffer is also influenced by the Prefetch factor (-prefetchFactor) on page 180 parameter. See the *OpenEdge Data Management: Database Administration* for a discussion of the interaction between the parameters.

Prefetch Priority (-prefetchPriority)

Use Prefetch Priority (`-preFetchPriority`) to utilize a "pollskip" that adds *`n`* records to the network message of an in-process prefetch query without additional polling.

Operating system and syntax	**UNIX / Windows**	**`-prefetchPriority n`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	32766	0	—	0[25]

`n`

The number of prefetch records to add to a network message.

Adding more than one record to the message at a time can greatly reduce the number of `poll()` calls being made on a large database configuration. The expected result is that reducing `poll()` calls improves overall system performance by decreasing the "System CPU" cycles allow for increased "User CPU" cycles. The pollskip only occurs if the previous `poll()` call returned nothing waiting.

Preload CLR (-preloadCLR)

Use Preload CLR (`-preloadCLR`) to load the .NET Common Language Runtime (CLR) and any specified assemblies into the ABL session at startup. This provides for deterministic loading of the CLR and assemblies when the application is started.

Operating system and syntax	**Windows**	**`-preloadCLR`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

If you do not specify -`preloadCLR`, the AVM loads the CLR the first time ABL refers to, or uses, a .NET type or object either at compile time or runtime. The individual assemblies are loaded as needed.

[25] The default of 0 produces the pre-Release 11.1 behavior.

If you specify `-preloadCLR` and the client machine does not have an appropriate version of the .NET framework installed, OpenEdge raises an error and shuts down.

For more information on accessing .NET objects using the ABL, see *OpenEdge Development: GUI for .NET Programming.* For information on the .NET CLR, refer to the Microsoft .NET Framework documentation.

Configuration Properties File (-properties)

Use Configuration Properties File (`-properties`) to identify the properties file that the AdminServer uses internally to specify startup parameters when starting a database server or servergroup.

Operating system and syntax	**UNIX / Windows**	**`-properties` *`filename`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	—	—	—	—

`filename`

The name of the properties file.

The default is `$DLC/properties/conmgr.properties`.This is an internal parameter, you should not use this parameter directly.

Proxy Host (-proxyhost)

Use the Proxy Host (`-proxyhost`) parameter to specify the name of the host or the IP address of the host at which the HTTP-based proxy server is located.

Operating system and syntax	**UNIX / Windows**	**`-proxyhost` { *`host-name`* | *`IP-address`* }**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Connection	—	—	—	—

`host-name`

The name of the host at which the HTTP-based proxy server is located.

`IP-address`

The IP address of the host at which the HTTP-based proxy server is located.

Note: If you specify the Proxy Host (-`proxyhost`) parameter, you must also specify the Proxy Port (-proxyport) on page 184 parameter.

When binding to a WSDL file via the server object's CONNECT( ) method, all connections made using the HTTP or HTTPS protocol connect to the proxy server at the specified host and port.

When you specify this parameter, all connections made by an AppServer client using the HTTP or HTTPS protocol connect to an AppServer Internet Adapter (AIA) instance using the proxy server at the specified host and port.

Proxy Password (-proxyPassword)

Use Proxy Password (`-proxyPassword`) when `-proxyhost` and `-proxyport` are specified and the Proxy server requires authentication.

Operating system and syntax	**UNIX / Windows**	**`-proxyPassword` *`password`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

`password`

Password used to authenticate the AppServer client to the HTTP-based Proxy server. *`password`* can be a string of up to 512 printable ASCII characters.

You can access `-proxyPassword` using the `SESSION:PROXY-PASSWORD` attribute.

If `-proxyhost` and `-proxyport` are not specified on the command line to the OpenEdge client, any value for `-proxyPassword`, whether specified with the startup parameter or with the `SESSION` attribute, is ignored.

`password` is validated during the AppServer `CONNECT()` method. If the *`password`* is invalid, the `CONNECT()` method fails and the AVM issues an error message.

If `-proxyUserid` is not specified or `SESSION:PROXY-USERID` is not set, the `CONNECT()` method ignores any value for `-proxyPassword`.

Proxy Port (-proxyport)

Use the Proxy Port (-`proxyport`) parameter to specify the port on which the HTTP-based Proxy server is listening.

Operating system and syntax	UNIX / Windows	`-proxyport { port-number }`		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Connection	65536	1	—	—

`port-number`

The port number of the HTTP-based proxy server.

When you specify this parameter, all connections made by an AppServer client using the HTTP or HTTPS protocol connect to an AppServer Internet Adapter (AIA) instance using the proxy server at the specified host and port.

When binding to a WSDL file via the server object's `CONNECT( )` method, all connections made using the HTTP or HTTPS protocol connect to the proxy server at the specified host and port.

Note: If you specify the Proxy Port (-`proxyport`) parameter, you must also specify the Proxy Host (-proxyhost) on page 182 parameter.

Proxy Userid (-proxyUserid)

Use the Proxy Userid (-`proxyUserid`) to authenticate an AppServer client to the HTTP-based Proxy server.

Operating system and syntax	UNIX / Windows	`-proxyUserid user-id`		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

`user-id`

`user-id` used to authenticate an AppServer client to the HTTP-based proxy server. The *`user-id`* can be a string of up to 512 printable ASCII characters, including the space character.

You can access -proxyUserid using the SESSION:PROXY-USERID attribute.

If -proxyhost and -proxyport are not specified on the command-line to the OpenEdge client, any value for the -proxyUserid, whether specified with the startup parameter or with the SESSION attribute, is ignored.

The *user-id* is validated during the AppServer CONNECT() method. If *user-id* is invalid, the CONNECT() method fails and the AVM issues an error message.

If -proxyPassword is not specified or SESSION:PROXY-PASSWORD is not set, and -proxyUserid is specified, a blank proxy password is used by the AppServer CONNECT() method.

Quick Request (-q)

Use Quick Request (-q) to tell the AVM to search PROPATH directories only on the first use of a procedure.

Operating system and syntax	**UNIX / Windows**	**-q**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

Ordinarily in an ABL procedure, when the RUN statement is used to run a subprocedure, the AVM searches the directories named by the PROPATH environment variable, looking for a procedure of the same name with a .r extension. If it finds a file with a .r extension (an r-code file), it checks to make sure the r-code file has not changed since that r-code file was created.

This search is very useful in a development environment where procedures change regularly and you want to make sure you are always running the most current version of your application. However, in a production environment, you might want to bypass this search.

With Quick Request (-q), after the initial search, if the procedure still resides in memory or in the local session-compiled file, the AVM uses that version of the procedure rather than searching the directories again. However, the AVM always checks whether Data Dictionary definitions related to a procedure were modified. If they were modified, the AVM displays an error when it tries to retrieve the procedure.

Non-reliable I/O (-r)

Use Non-reliable I/O (`-r`) to enable buffered I/O to the before-image file. In most cases, avoid using this parameter because it might put database integrity at risk.

Operating system and syntax	**UNIX / Windows**	**`-r`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Connection	—	—	Unbuffered I/O	Unbuffered I/O

Caution: If you run OpenEdge with the `-r` parameter and your system fails because of a system crash or power failure, you cannot recover the database. You must restore the database from a backup and restart processing from the beginning. If you are updating a database using buffered I/O and there is an OpenEdge system failure, but no hardware failure, OpenEdge can recover the database.

In general, you always want complete database integrity or you want fast performance for large tasks, such as loading large amounts of data or doing fiscal year-end processing. The `-r` parameter does not guarantee complete database integrity, and No Crash Protection (-i) on page 120 is faster in terms of performance.

Alternate Random Number Generator (-rand)

Use Alternate Random Number Generator (`-rand`) to specify whether to use the original random number generator or the alternate one.

Operating system and syntax	**UNIX / Windows**	**`-rand` *`n`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	2	1	1	1

`n`

> The type of random number generator. A value of 1 (the default) indicates that OpenEdge should use the original generator; specify a value of 2 to use the alternate.

The original number generator always generates the same random sequence; that is, the numbers that it generates are random, but each time a session starts, it gives you the same set of numbers from the last session. If you have to generate a different sequence of random numbers, specify the alternate generator. This generator returns a number from a **pseudorandom** sequence of numbers rather than a truly random sequence.

Require Username (-requireusername)

Use Require Username (`-requireusername`) to require a username and password from the `ubroker.properties` file for each user starting servers of the AdminServer (AppServer, SonicMQ, and WebSpeed).

Operating system and syntax	**UNIX / Windows**	**`-requireusername`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Connection	—	—	—	—

This parameter ensures that only valid users are able to start an AdminServer.

Reread Fields (-rereadfields)

Use Reread Fields (`-rereadfields`) to suppress inadvertent runtime errors that result from using field lists in ABL code. Instead of generating an error message, the AVM ignores the field list and fetches the entire record.

Operating system and syntax	**UNIX / Windows**	**`-rereadfields`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

Field lists restrict the fields returned from a record query. They allow you to retrieve only specified fields rather than all the fields of a record. Field lists are implemented with the FIELDS or EXCEPT options in FOR, DO PRESELECT, and DEFINE QUERY statements. They can improve performance over a network by enabling more records to fit into a single network message.

Field lists can cause unexpected errors, particularly when you update existing code. It is possible, for example, that you may have eliminated a field that will be required by a new or revised subroutine or trigger. The result will be a runtime error that may be difficult to debug.

You can use the `-rereadfields` startup option if you have unexpected runtime errors resulting from field lists. When an error occurs, the `-rereadfields` startup option causes the AVM to ignore the field list and to fetch the entire record.

Also note the difference between `-rereadfields` and `-fldisable`. The Field List Disable (`-fldisable`) option causes the AVM to ignore **all** field lists and to fetch the entire record for every query. The Reread Field List (`-rereadfields`) option causes the AVM to ignore a particular field list and fetch the entire record only when an error occurs due to a missing field. Therefore, using `-rereadfields` is likely to have less of a negative impact on performance.

Note:

- You should not use `-rereadfields` during actual application development since you will not see the error messages that could help you determine what fields are required in a field list.
- Also keep in mind that the compiler does not search for all references to a record to ensure that required fields are available. Therefore, it is better to use a field list only when there is some serious performance problem retrieving the entire record, when use of the record is not complicated, and when references to fields are predictable.

For more information about using field lists, see *OpenEdge Getting Started: ABL Essentials*.

Reread Nolock (-rereadnolock)

Use Reread Nolock (`-rereadnolock`) to tell the AVM how to decide between multiple copies of a cached record that were read from the database using `NO-LOCK`. By default, the AVM chooses the oldest of the `NO-LOCK` copies. When you start a session with `-rereadnolock`, the AVM chooses the most recently cached `NO-LOCK` copy.

Operating system and syntax	**UNIX / Windows**	**`-rereadnolock`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

The AVM stores records read from the database in internal buffers that are linked to the record buffers of your ABL application. When two internal buffers for the same record exist, the AVM decides which to keep and use by rules based on the locking level and age of each copy. This parameter only affects how the decision is made when both of the following conditions are true:

- A record is read from the database with `NO-LOCK`.
- The AVM finds an older `NO-LOCK` copy of that same record already in memory.

For more information on record locking, see the chapter on data handling and record locking in *OpenEdge Getting Started: ABL Essentials*.

Note: This parameter causes no extra database activity; it simply determines whether to keep the older or newer of the copies already in memory.

You use this parameter to resolve client-server currency conflicts. You also use it to resolve server-to-server currency conflicts by using it as an AppServer startup parameter through the OpenEdge Management or OpenEdge Explorer, or by setting the `srvrStartupParam` property in the `ubroker.properties` file for the appropriate AppServer. When applications use record buffers with a large scope, such as default buffers scoped to the main block of long life persistent procedures as you might have on an AppServer, using `-rereadnolock` is strongly advised.

Remember the following when using `-rereadnolock`:

- It has no affect on records that are retrieved through RECID or ROWID. In that case, the AVM will not reread the record. Instead, it uses the copy of the record already stored in the buffer. If you need the most current version of the record, use the RELEASE statement on all buffers that contain a copy of the record before reading the record, or use the FIND CURRENT or GET CURRENT statement to reread the record.
- It has no affect on the behavior of the query cache used for a query with NO-LOCK that is specified through the CACHE phrase of the DEFINE QUERY statement. To force the AVM to always re-read the record, set the cache size to zero (0). However, this may significantly degrade performance if the database is accessed across a network. Set the cache size to zero only when it is critical to retrieve the most current version of a record.
- It has no affect on the behavior of the prefetch cache that is used by default when retrieving records NO-LOCK across the network. By default, when executing a CAN-FIND function or a FIND, FOR, or OPEN QUERY statement on a database that is accessed across a network, the AVM fetches several records at a time and stores them in a prefetch cache. The AVM will only send a request to the database server to fetch more records if the requested record is not in the current prefetch cache. If the record is in the current cache, the AVM will not read a new copy of that record even if -rereadnolock is set. To eliminate this cache so that the most current version of the record is always read, use the NO-PREFETCH keyword in the appropriate statements. However, using the NO-PREFETCH keyword may significantly degrade performance. Set NO-PREFETCH only if it is critical to retrieve the most current version of a record.

Re-usable Objects Cache (-reusableObjects)

Use Re-usable Objects Cache (`-reusableObjects`) to specify the number of deleted class objects that the AVM stores for later re-initialization.

Operating system and syntax	**UNIX / Windows**	**`-reusableObjects` *`cache-size`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	2147483647	0	25	25

`cache-size`

Number of deleted class objects to store for later re-initialization.

This parameter can improve the performance of applications that use ABL classes. When a class object is deleted, either by the DELETE OBJECT statement or through garbage collection, the AVM caches the object. When the AVM instantiates the same class later, the stored object is re-initialized and removed from the cache. This re-use saves much of the overhead of instantiating a class.

Note: The cache only stores class-based objects. However, it does not store .NET classes, .NET-derived ABL classes, classes with STATIC elements, or session-compiled classes.

The cached class objects continue to hold some resources. You might want to modify the settings of Directory Size (-D) on page 100 and Maximum Memory (-mmax) on page 150 to account for the number of cached objects. For more information on the re-use of objects, see the DELETE OBJECT statement in *OpenEdge Development: ABL Reference* and the section on object life-cycle in *OpenEdge Development: Object-oriented Programming*.

Run ABL Client (-rg)

Use Run ABL Client (`-rg`) to indicate that you want to run the ABL client.

Operating system and syntax	UNIX / Windows	**`-rg`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

Use this parameter if you have more than one type of OpenEdge client and want to limit access/functionally of the session to the ABL client.

Read-only Media (-RO)

Use Read-only (`-RO`) to open a database for read-only access or to access a database stored on read-only media (a read-only optical disk, for example).

Operating system and syntax	UNIX / Windows	**`-RO`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Connection	—	—	—	—

Note: Do not run servers for a read-only database. OpenEdge does not support servers for read-only users.

If you use the `-RO` parameter when other users are updating the database, you might see invalid data, such as stale data or index entries pointing to records that have been deleted.

A read-only session is essentially a single-user session. Read-only users do not share database resources (database buffers, lock table, index cursors). However, unlike a normal single-user database, a read-only database can be simultaneously accessed by multiple `-RO` users.

When a read-only session starts, it does not check for the existence of a lock file for the database. Furthermore, a read-only user opens the database file, but not the log or before-image files. Therefore, read-only user activity does not appear in the log file.

If a database is opened with `-RO`, the string returned by the `DBRESTRICTIONS` function includes the keyword `READ-ONLY`.

Run Query Client (-rq)

Use Run Query Client (`-rq`) to indicate that you want to run the Query client.

Operating system and syntax	**UNIX / Windows**	**`-rq`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

Use this parameter if you have more than one type of OpenEdge client and want to limit access or functionality of the session to the Query client.

Run Run-time Client (-rr)

Use Run Run-time Client (`-rr`) to run the Run-time Client.

Operating system and syntax	**UNIX / Windows**	**`-rr`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

Use this parameter if you have more than one type of OpenEdge client and want to limit access or functionality of the session to the Run-time Client.

Encrypted Compiler Mode (-rx)

Use Encrypted Compiler Mode (`-rx`) to enable query or run-time OpenEdge clients to compile encrypted source code and access the Data Dictionary to manage schema (for example: for security management and to dump or load `.df` files).

Operating system and syntax	UNIX / Windows	**-rx**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

If OpenEdge is started with `-rx`, you can only compile encrypted source code. Use `-rx` when you are shipping an application in the form of encrypted source code to end-user sites having query or run-time versions of OpenEdge. You must prepare a tool that end users can run to compile your code, and this tool must include a command that invokes OpenEdge with `-rx`.

Service Name (-S)

Use Service Name (`-S`) to specify the service or port number to be used when connecting to a broker process or used by a broker process on the host machine.

Operating system and syntax	UNIX / Windows	-S { ***service-name*** | ***port-number*** }		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Connection, Database Server	—	—	—	—

service-name

The name of the service to be used by the broker process.

port-number

The port number of the host; if using OpenEdge Management or OpenEdge Explorer, the port number of the NameServer.

You must use this parameter when starting:

- A broker or server on a machine that will serve remote users
- Multi-user OpenEdge session as a remote user

The system administrator must make an entry in the `services` file that specifies the server or broker name and port number.

When the broker spawns a server, the server inherits all of the network parameters (except the Service Name parameter) from the broker. Because there is no restriction on the number of brokers you can start, you can have multiple brokers running with different network parameters. See also the Server Group (-servergroup) on page 196 startup parameter description.

The following table shows how the broker, server, and remote client interpret each of their parameters when you use the `-S` parameter.

Table 5: Parameter interpretation with Service Name (-S)

Module	Interpretation
Broker	Parameters apply to the connections on which the broker is listening for connection requests from remote clients
Server	Parameters apply to the connection between the server and the remote client
Remote Client	Parameters identify the connection parameters to the broker or the server

To run multi-user OpenEdge from a remote network node, use both the Host Name (-H) on page 118 and Service Name (-S) parameters.

Stack Size (-s)

Use Stack Size (`-s`) to change the size of the stack (an internal memory area used by ABL program modules).

Operating system and syntax	UNIX / Windows	**-s *n***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	5,242,878[26]	9	40	40

n

The size of the stack in 1KB units.

Note: It is recommended that the `-s` session startup parameter be set to 128. Setting this startup parameter to a smaller value may result in runtime errors in some environments.

[26] Actual maximum value is limited only by available memory

Increase the stack size if one of the following messages appears:

```
SYSTEM ERROR: stkpush: stack overflow. Increase -s parameter.
SYSTEM ERROR: stkditem: stack overflow. Increase -s parameter.
```

Stack overflow errors are most likely when data definitions are loaded for very large tables or use recursive procedures.

Note: If your application passes array parameters to procedures or functions, your stack size requirements can increase significantly.

Schema Lock Wait Queue (-schlockwq)

Use Schema Lock Wait Queue (`–schlockwq`) to alter the schema locking algorithm.

Operating system and syntax	**UNIX**	**-schlockwq**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	—	—	—	3

When the database broker is started with `–schlockwq`, the schema lock scheduler forces a shared schema lock request to wait when an exclusive schema lock request is queued. Once an exclusive schema lock request is queued, it appears to all subsequent schema lock requests that an exclusive schema lock has been granted (meaning all subsequent schema lock requests will queue). Without `–schlockwq`, shared locks continue to be granted, even though an exclusive lock is queued.

With `–schlockwq`, wake up of users waiting for a schema lock is performed in the order they were received. Without `–schlockwq`, all users are awoken whenever any schema lock conflict is resolved, causing a race to see which user gets the available lock. When running with `–schlockwq` enabled at broker startup, only those users allowed to get the schema lock will be awoken.

All existing lock strength conflicts and properties are unchanged, as follows:

- Exclusive locks conflict with shared and exclusive locks for other users.
- Share locks may coexist with other share locks but conflict with exclusive locks of other users.
- An individual user may possess at most one shared lock and one exclusive lock concurrently.
- An individual user may never possess more than one lock of the same lock strength (i.e. two concurrent share locks for the same user is prohibited).
- An individual user may only ever have one entry on the wait queue.
- These behaviors are inherently supported by each of the language engines and enforced by the storage engine.

The `–schlockwq` parameter has no impact when the database is started in single-user mode.

Screen-value Mode (-scrvalmode)

Use Screen-value Mode (`-scrvalmode`) to specify the mode in which the SCREEN-VALUE attribute indicates that the selected item in a combo-box item list is empty.

Operating system and syntax	**Windows**	**`-scrvalmode n`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	0	0

n

The screen-value mode. Valid modes are:

- **1** — If the selected item in the combo-box list is empty, the SCREEN-VALUE attribute for the combo-box returns the empty string (""). Note that if no item in the combo-box list is selected, the SCREEN-VALUE attribute returns the `Unknown value (?)`. In either case, the INPUT-VALUE attribute returns the empty string ("").
- **3** — The same as mode 1, with the following exception: if the SCREEN-VALUE attribute for the combo-box returns the `Unknown value (?)`, the INPUT-VALUE attribute also returns the `Unknown value (?)`.

Prior to OpenEdge R10.0, the SCREEN-VALUE attribute for a combo-box returns the empty string ("") if the selected item in the list is empty. Starting with Release 10.0, the SCREEN-VALUE attribute for a combo-box returns the `Unknown value (?)` if the selected item in the list is empty. In either case, if no item in the combo-box list is selected, the SCREEN-VALUE attribute returns the `Unknown value (?)`.

This startup parameter does not apply to combo-box browse columns.

Semaphore Sets (-semsets)

Use Semaphore Sets (`-semsets`) to change the number of semaphore sets available to the OpenEdge broker.

Operating system and syntax	**UNIX**	**`-semsets n`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	Maximum Number of Users +1	1	—	3

n

An INTEGER specifying the number of semaphore sets available to the OpenEdge broker.

When more than 1,000 users connect to a single database, there might be high contention for the semaphore set. If there is a lot of semaphore contention on a system, using multiple semaphore sets helps alleviate this contention and improve performance on high user counts.

For more information on using semaphore sets, see *OpenEdge Data Management: Database Administration*.

Server Group (-servergroup)

Use Server Group (-servergroup) to identify the logical collection of server processes to start.

Operating system and syntax	**UNIX / Windows**	**-servergroup *name***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	—	—	—	—

name

Specifies the name of the logical collection of server processes.

The *name* you specify must match the name of a servergroup in the conmgr.properties file. Use the OpenEdge Management or OpenEdge Explorer tool to create servergroups and save them in the conmgr.properties file. Do not edit the conmgr.properties file directly. To start a database configuration use OpenEdge Management, OpenEdge Explorer, or the DBMAN utility.

Note: You can use the mergeprop utility installed with OpenEdge to edit the conmgr.properties file. For information on using mergeprop, see *OpenEdge Getting Started: Installation and Configuration*.

Type of Server to Start (-ServerType)

Use the Type of Server to Start (`-ServerType`) parameter to limit the type of server the broker can start.

Operating system and syntax	UNIX / Windows	**`-ServerType [4GL	SQL	Both]`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database ServerSQL	—	—	Both	Both

`4GL`

Indicates the broker supports only OpenEdge servers.

`SQL`

Indicates the broker supports only SQL servers.

`Both`

Indicates the broker supports both OpenEdge and SQL servers.

To change the value of `-ServerType`, use the `config.properites` file or the command line.

Session Timeout (-sessiontimeout)

Use Session Timeout (`-sessiontimeout`) to change the length of time that a SSL session will be cached.

Operating system and syntax	UNIX / Windows	**`-sessiontimeout`** ***`n`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	—	—	—	180

`n`

Specifies in seconds the length of time a SSL session will be held in the SSL session cache. The default is 180 seconds.

SSL session caching allows a client to reuse a previously established SSL session if it reconnects prior to the SSL session cache timeout expiring.

Shared memory segment size (-shmsegsize)

Use Shared memory segment size (`-shmsegsize`) to specify the size of the largest shared memory segment the server can allocate.

Operating system and syntax	**UNIX / Windows**	**`-shmsegsize` *`n`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	System dependent[27]	128 for 32-bit systems, 1024 for 64-bit systems.	—	Configuration and system dependent.[28]

`n`

The maximum number of bytes, specified in MB or GB, in a shared memory segment.

The following table lists the supported values by platform.

Table 6: Values for -shmsegsize

Platform	Values in MB	Values in GB[29]
32-bit platforms	128	—
	256	—
	512	—
	1024	1g
	2048	2g
	4096	4g

[28] If not specified, the first attempt is to allocate one segment for all the required memory. If the required memory exceeds the operating system maximum segment size, the system maximum size is used.

[27] Limited by available memory.

[29] **You must include the "g" if you specify the value in gigabytes.**

Platform	Values in MB	Values in GB[29]
64-bit platforms	1024	1g
	2048	2g
	4096	4g
	8192	8g
	16384	16g
	32768	32g

The shared memory segment size can impact performance. Increasing the segment size decreases the number of segments allocated and reduces the resources needed to manage the segments. See *OpenEdge Data Management: Database Administration* for more information.

Show PPU Error (-showppuerr)

Use Show PPU Error (-`showppuerr`) to restore the warning message `4132` ("Invalid character unit value `<N>`. Changed to `320`.")

Note: Starting with Release 10.1B, this startup parameter has no effect; it is supported only for backward compatibility.

Operating system and syntax	UNIX / Windows	`-showppuerr`		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

Prior to Release 10.1B, the AVM and the compiler raise warning message `4132` when an ABL program tries to set a column or width attribute to a value greater than `320`. The AVM or the compiler resets the column or width attribute value to `320` and suppresses the associated warning message.

Starting with Release 10.1B:

- At compile time, the compiler raises warning message `4132` when an ABL program tries to set a column or width attribute to a value greater than 320. The AVM resets the column or width attribute value to `320` and the compiler generates the associated warning message (for example: `Invalid character unit value 200,000. Changed to 320.`).
- At run time, the AVM raises warning message `13734` ("Invalid character unit value `%|<value>`. Changed to `%|<max>.`") when an ABL program tries to set a column or width attribute to a

[29] You must include the "g" if you specify the value in gigabytes.

value greater than 131,072. The AVM resets the column or width attribute value to `131,072` and the AVM generates the associated warning message (for example: `Invalid character unit value 200,000. Changed to 131,072.`).

Note: The AVM stores column and width values in the internal storage format called Progress Portability Units (PPU).

Spin Lock Retries (-spin)

Use Spin Lock Retries (`-spin`) to set a value to use the spin lock mechanism or a system of semaphores and queues.

Operating system and syntax	**UNIX / Windows**	**`-spin` *`n`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	—	0	0	10,000, or 6000 * # CPUs[30]

`n`

The number of times a process tries to acquire a latch before pausing.

If the value of *`n`* is greater than 0, then a spin lock algorithm is used for shared-memory data structures. When a process has to lock a shared-memory structure, the process tries up to *`n`* times to acquire the latch for that structure. If it has not acquired the latch in *`n`* tries, the process pauses, or naps. The length of the pause increases gradually if the process repeatedly fails to acquire a latch. After the allotted nap time, the process wakes up and tries to acquire the lock again. If it fails to acquire the lock, it will retry up to the number of tries specified by *`n`*.

If the value of *`n`* is 0, a system of semaphores and queues is used instead of spin locks. The spin lock algorithm is much more efficient than using semaphores when you have multiple processors.

On multiprocessor machines, the default value is 6000 * # of CPUs. If this causes too much CPU usage, reduce the value. If you have fast processors, a value as high as 10,000 * # of CPUs might be effective.

You can evaluate the `-spin` values through the OpenEdge Monitor (PROMON utility) R&D options. See *OpenEdge Data Management: Database Administration* for more information on the PROMON utility. The `-spin` parameter controls the performance indicator called resource waits. By setting the `-spin` value higher, you can reduce the resource waits. Note that when setting the `-spin` value higher ceases to reduce the resource waits, continuing to set it higher can adversely effect CPU utilization. To view the resource waits value:

- Access PROMON and enter `R&D` at the main menu.

[30] For single CPU systems, the default is 10,000. For multi-CPU systems, the default is 6000 * # of CPUs

- Choose option **3. Other Displays**, then choose option **1. Performance Indicators** to view the resource waits. Resource waits is the last item reported in the listing.

SQL Open Cursors (-SQLCursors)

Use SQL Open Cursors (`-SQLCursors`) to define the maximum number of cursors open at any one time.

Operating system and syntax	**UNIX / Windows**	**`-SQLCursors` *`value`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
SQL	512	10	—	50

`value`

Number of cursors open at one time.

SQL rowid identifier(-SQLQuotedRowid)

Use SQL rowid identifier (`-SQLQuotedRowid`) to specify how the quoted string "rowid" is interpreted.

Operating system and syntax	**UNIX / Windows**	**`-SQLQuotedRowid` [`std` | *`value`*]**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	—	—	std	std

`std` | *`value`*

Specify behavior of the string "rowid":

- **`std`** — specify `std` to indicate that all instances of the quoted string "rowid" are interpreted as an SQL identifier (such as a column name).
- ***`value`*** — specify any string other than `std` to indicate that all instances of rowid, quoted or not, are interpreted as the SQL rowid function.

Example

Start the database server as shown:

```
mprosrv myDB -S  1234 -H myhost  -SQLQuotedRowid std
```

After starting the server and connecting a client to the database, query a column named rowid as shown:

```
select custnum, "rowid"  from pub.my_customer where "rowid" = 10 ;
```

Query both the column named rowid (quoted), and the value of the rowid function (no quotes) as shown:

```
select custnum, "rowid", rowid from pub.my_customer where "rowid" = 10 ;
```

SQL Stack Size (-SQLStack)

Use SQL Stack Size (`-SQLStack`) to change the size of the SQL stack (an internal memory area used by SQL program modules). Use the command line to change the value of `-SQLStack`.

Operating system and syntax	**UNIX / Windows**	**`-SQLStack` *`n`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
SQL	System Dependent	1	1	1

`n`

The size of the stack in 1KB units.

SQL Statement Cache Size (-SQLStmtCache)

Use SQL Statement Cache Size (`-SQLStmtCache`) to set the number of statements that can be stored in the SQL cache.

Operating system and syntax	**UNIX / Windows**	**`-SQLStmtCache` *`n`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
SQL	—	—	—	100

`n`

The number of statements to be stored in the SQL Statement Cache.

Use the command line to change the value of `-SQLStmtCache`.

SQL Sorting Memory (-SQLTempBuff)

Use SQL Sorting Memory (`-SQLTempBuff`) to define the amount of memory to use as a data buffer for each `SORT` operation.

Operating system and syntax	**UNIX / Windows**	**`-SQLTempBuff` *`value`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
SQL	—	8	1000	1000

`value`

Size in K-bytes of memory to use for each `SORT` operation.

This is a performance-tuning parameter and is not ordinarily required.

Note: Each connection allocates one temporary table buffer.

SSL-based Connection (-ssl)

Use SSL-based Connection (`-ssl`) to specify a Secure Sockets Layer (SSL) connection to all database and client connections for data privacy.

Operating system and syntax	UNIX / Windows	**-ssl**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Connection, Database Server	—	—	—	—

SSL provides an authenticated and encrypted peer-to-peer TCP/IP connection.

Note: Be sure you need SSL before using this parameter. SSL incurs more or less heavy performance penalties, depending on resources and load.

For more information, see *OpenEdge Getting Started: Core Business Services - Security and Auditing*.

Strict Entity Resolution (-strictEntityResolution)

Use Strict Entity Resolution (`-strictEntityResolution`) to control whether or not the XML parser (DOM or SAX) will attempt to resolve an external entity if that entity is located outside of the directories in the `SCHEMA-PATH` attribute of any given `SAX-reader` or `X-document` object handle or `XML-SCHEMA-PATH` attribute of the `WEB-OBJECT` system handle.

Operating system and syntax	UNIX / Windows	**-strictEntityResolution *n***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	1	1

n

Valid values are 0 for non-strict entity resolution, and 1 for strict entity resolution. If the parameter is not specified, the default value is 1.

In strict entity resolution mode, the parser (DOM or SAX) will NOT attempt to resolve an external entity in a location not in the `SCHEMA-PATH` or `XML-SCHEMA-PATH` attribute. In non-strict mode, the parser will attempt to resolve an external entity. Using strict entity resolution mode can help prevent an external caller from gaining access to data located in directories not included in the `SCHEMA-PATH` or `XML-SCHEMA-PATH` attribute.

This behavior can also be controlled on a case-by-case basis by the `STRICT-ENTITY-RESOLUTION` attribute of a particular `SAX-reader` or `X-document` object handle or the `XML-STRICT-ENTITY-RESOLUTION` of the `WEB-OBJECT` system handle. Setting the `STRICT-ENTITY-RESOLUTION` or `XML-STRICT-ENTITY-RESOLUTION` attribute overrides the behavior indicated by the startup parameter. See *OpenEdge Development: ABL Reference* for more information.

Stash Area (-stsh)

Use Stash Area (`-stsh`) to specify the number of 1KB blocks to allocate to the stash area, which OpenEdge uses as temporary storage for modified index fields.

Operating system and syntax	**UNIX / Windows**	**-stsh *n***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	31	1	2	2

n

The number of 1KB blocks for the stash area.

The default size is sufficient for most index fields, but large character fields with a word index might require more space.

You can calculate an appropriate stash size by taking the size in bytes of the largest field with a word index, doubling that value, and adding 1KB. For example, if the largest word indexed field is 2,000 characters long, set `-stsh` to 5 to create a stash area of 5KB. If you do not allocate enough space, OpenEdge terminates with a fatal error. If this occurs, restart OpenEdge with a larger stash area.

Temporary Directory (-T)

Use Temporary Directory (`-T`) to specify a directory for temporary files.

Operating system and syntax	UNIX / Windows	**-T *dirname***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session, Database Server	—	—	Working directory	Working directory

dirname

The directory in which to create temporary files.

If you do not use this parameter, OpenEdge creates temporary files in the current working directory.

The database broker passes the setting of `-T` to any servers it spawns. Manual servers and secondary brokers can specify their own `-T` or take the default; they do not inherit the setting from the primary broker. Remote Client connections can also specify a `-T` setting different from the server it is connecting to. Temporary space needed in the remote client is stored in its own `-T` directory. Temporary space needed in the server to process a request for a remote client is stored in the server's `-T` directory.

On UNIX and in Windows, these filenames begin with one of the prefixes listed in the table below, and are followed by a number.

Note: Access the `-T` parameter using the `SESSION:TEMP-DIR` attribute. This attribute is read-only.

The following table shows descriptions of these prefixes:

Temp file	Usage
`bcf`	Internal file for tracking recid changes
`dbg`	Debugger listing file
`dbi`	Stores temporary tables
`lbi`	Local before-image (subtransaction undo)
`lst`	Temporary file used by the compile-listing feature of the COMPILE statement
`obj`	Temporary `.r` file
`*.ped`	Edit buffer contents. `ped` is NOT a prefix, it is a file extension.

Temp file	Usage
`rcd`	Cache of r-code being run in a session
`rpf`	Raw data used by the Profiler
`srt`	Temporary sort space
`trp`	Stores Data Dictionary changes until they are saved

On UNIX, OpenEdge names these files uniquely for each user to avoid filename conflicts. Furthermore, OpenEdge stores these files as "unlinked" with no visible name in the UNIX file system, unless you use Save Temp Files (`-t`).

In Windows, if two or more users share the same working directory and there is a conflict of temporary files, the following message appears:

```
Unable to open or create filename, error 3
```

Save Temp Files (-t)

Use Save Temp Files (`-t`) to make temporary files visible on UNIX.

Operating system and syntax	UNIX	**-t**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

Typically, temporary files (`LBI`*`number`*, `SRT`*`number`*, `PGE`*`number`*, `DBI`*`number`*, `RCD`*`number`*, and `TRP`*`number`*) are invisible (unlinked on UNIX) when OpenEdge is running. In these filenames, *`number`* is the process ID of the user's OpenEdge session. The `-t` parameter also makes the entries visible after OpenEdge aborts. However, if OpenEdge aborts, delete the temporary files yourself.

With or without the `-t` parameter, OpenEdge always deletes the temporary files when it terminates normally.

If you are running a Windows ABL client against a database with transparent data encryption enabled, be aware that encrypted table data is stored in decrypted form inside the OpenEdge temporary files. Protect the OpenEdge temporary files, as well as any application-generated temporary files, using operating system security such as built-in file encryption or a third party products.

You cannot use Save Temp Files (`-t`) with an encryption-enabled database. If you try to start an encryption-enabled database with -t, the following error occurs:

```
Connection to an encrypted database is not permitted with -t. (15389)
```

Table consistency check (-TableCheck)

Use Table consistency check to enable consistency checking for all record blocks of the specified table or table partition (excluding BLOBS).

Operating system and syntax	**UNIX / Windows**	**`-TableCheck` { *`tablename`* |** ***`ownername.tablename`* |** ***`ownername.tablename.partitionname`*}**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Connection, Database Server	—	—	—	—

`ownername`

> Specifies the owner of the table you want to check. You must specify an owner name unless the table's name is unique within the database, or the table is owned by PUB. By default, ABL tables are owned by PUB.

`tablename`

> Specifies the name of the table for consistency checking.

`partitionname`

> In a database with partitioned tables, specifies a named partition. Specify the composite initial partition by using `initial` as *`partitionname`*.

Table consistency check validates a block is still physically correct after an operation has been performed.

You can only specify one table or partition with `-TableCheck`. To check multiple partitions, you must check the entire table. To check multiple tables, you must apply consistency checking to the entire database with Database consistency check (-DbCheck) on page 103, or if multiple tables are in one area, you can apply consistency checking to an entire area with Area consistency check (-AreaCheck) on page 70.

When specifying the table or partition to check, the following parsing rules are applied:

- If one variable is supplied, it is interpreted as a table name.

- If two variables are supplied, they are interpreted as the owner and table name (`ownername.tablename`).
- Three variables **must** be supplied to specify a table partition (`ownername.tablename.partitionname`).

Table consistency checking can be enabled or disabled while your database is online with PROMON. See *OpenEdge Data Management: Database Administration* for more information.

Table consistency check can be enabled for a single user client or for any offline utility. If the utility is read-only, such as DBANALYS, the `-TableCheck` parameter is ignored. Online utilities determine whether or not to perform table consistency checking during execution based on the use of `-TableCheck` at broker startup or by the enablement/disablement of the table consistency checking in PROMON.

Table Range Size (-tablerangesize)

Use Table Range Size (`-tablerangesize`) to specify the number of tables for which you want to collect statistics.

Operating system and syntax	**UNIX / Windows**	**-tablerangesize *n***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	—	—	—	—

n

The number of tables for which you want to track access statistics.

For more information on tracking table statistics, see *OpenEdge Data Management: Database Administration*.

Speed Sort (-TB)

Use Speed Sort (`-TB`) primarily to improve sort performance, particularly during index rebuild operations.

Operating system and syntax	**UNIX / Windows**	**-TB *blocksize***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	31	1	8	8

blocksize

The block size in kilobytes, to allocate when sorting records for reports and when rebuilding indexes. The default block size is 8K but can be reset anywhere from 1KB to 31KB.

If you increase the block size beyond 8KB, OpenEdge sorts records faster but uses more memory and disk space in the process. If system memory is severely limited, you might actually choose to lower the block size.

Sort space is allocated in the `SRT` file, a temporary session file having a system-generated unique name with the prefix `srt`. OpenEdge uses the `SRT` file to store session-compile modules, to swap r-code segments into and out of the in-memory execution buffer, and as temporary work space during sorting (including rebuilding indexes). Use `-TB` to set the `SRT` file block size (that is, the increments by which the `SRT` file grows).

In summary, increase `-TB` to at least 8 to improve index rebuild operations by 15 percent or more. Or, decrease `-TB` to 1 to relieve severe Windows memory shortages during sort operations.

See also the Merge Number (-TM) on page 212, Maximum Memory (-mmax) on page 150, and PROLIB Swap (-pls) on page 178 startup parameters.

Time Stamp (-tstamp)

Use Time Stamp (`-tstamp`) to direct the AVM to use time stamp information rather than cyclic redundancy (CRC) values to enforce consistency between r-code and schema.

Operating system and syntax	**UNIX / Windows**	**`-tstamp`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Connection	—	—	—	—

Note: Progress Software Corporation recommends that you use CRC instead of time stamp.

When precompiling a procedure, the AVM inserts in the r-code file a list of referenced tables and their cyclic redundancy code values (CRCs) by default. At run time, OpenEdge compares the CRCs in the r-code to those in the database being run against. If the CRCs do not match, OpenEdge returns an error message.

Note:

- To use time stamp values rather than CRC values, specify Time Stamp (`-tstamp`) for the process that compiles the procedures.
- The Time Stamp (`-tstamp`) parameter only applies to a single database.

For more information on time stamp and CRC values, see *OpenEdge Deployment: Managing ABL Applications*.

Temp-table Schema Marshal (-ttmarshal)

Use Temp-table Schema Marshal (`-ttmarshal`) to specify the amount of schema information to marshal for temp-table parameters during an OpenEdge client session (that is, for an ABL client, a single WebSpeed agent, or single AppServer agent). The temp-table may be an independent temp-table or a member of a ProDataSet object.

Operating system and syntax	**UNIX / Windows**	**`-ttmarshal` *`n`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

`n`

A value between 0 and 5 representing the amount of schema information to marshal for temp-table parameters. The default value is 0 (which includes all schema information for all temp-table parameters).

The following table lists the possible schema marshaling parameter values.

Value	Description
0	Includes all schema information for all temp-table parameters
1	Minimizes schema information for static temp-table parameters
2	Minimizes schema information for all temp-table parameters
3	Excludes schema information for static temp-table parameters
4	Excludes schema information for all temp-table parameters
5	Excludes schema information for static temp-table parameters and minimizes schema information for dynamic temp-table parameters

You can override this startup parameter setting for an individual temp-table parameter by setting one of the following attributes on the `Temp-Table` object handle:

- `SCHEMA-MARSHAL` attribute
- `MIN-SCHEMA-MARSHAL` attribute
- `NO-SCHEMA-MARSHAL` attribute

Note: If you specify more than one of these attributes for an individual temp-table, the AVM uses the attribute you most recently specified. The `MIN-SCHEMA-MARSHAL` and `NO-SCHEMA-MARSHAL` attributes are supported only for backward compatibility. Use the `SCHEMA-MARSHAL` attribute instead.

For more information about these attributes, see *OpenEdge Development: ABL Reference*.

Merge Number (-TM)

Use Merge Number (-TM) to increase the speed of the merge phase of the sort process (at the cost of increased temporary memory usage).

Operating system and syntax	**UNIX / Windows**	**-TM *n***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	32	1	5	5

n

The number of blocks or streams to be simultaneously merged during the sort process.

See also the description of the Speed Sort (-TB) on page 209 startup parameter.

Temporary Table Database Block Size (-tmpbsize)

Use Temporary Table Database Block Size (-tmpbsize) to specify the temp-table block size.

Operating system and syntax	**UNIX / Windows**	**-tmpbsize *n***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	4	4

n

An INTEGER that specifies the temp-table database block size. Valid values are:

- **1** — Sets the block size to 1024 bytes.
- **4** — Sets the block size to 4096 bytes. This is the default value.
- **8** — Sets the block size to 8192 bytes.

The temp-table database block size dictates the size of each buffer in the temp-table database buffer pool for the current OpenEdge session.

Use Number of Buffers for Temporary Tables (-Bt) on page 81 to specify the number of buffers in the temp-table database buffer pool.

Note: With the changes made in Release 10.1B to provide support for expanded keys and temp-tables, the default block size for temp-tables has increased from a 1KB block size to a 4KB block size on all platforms. If you are creating very small temp-tables, you might notice a performance degradation with this change.

See *OpenEdge Getting Started: ABL Essentials* for more information on temp-tables.

Token (-tok)

Use Token (`-tok`) to specify the maximum number of tokens allowed in a single ABL statement.

Operating system and syntax	**UNIX / Windows**	**`-tok` *`n`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	2147483647	1	3000	3000

`n`

The number of tokens allowed in an ABL statement. The default is 3000 tokens.

If you exceed 3000 tokens in a single statement, the AVM displays a message advising you to increase the number of tokens per statement using this parameter.

Trigger Location (-trig)

Use Trigger Location (`-trig`) to identify the name of the directory or library that contains the database ABL triggers for a database.

Operating system and syntax	**UNIX / Windows**	**`-trig` { *`dir-name`* | *`lib-name`* }**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Connection	—	—	—	—

`dir-name, lib-name`

The location of the database ABL triggers.

Library names must have a `.pl` extension. the AVM interprets all other extensions as directory names. If you do not use Trigger Location (`-trig`) when connecting to the database, the AVM searches the `PROPATH` for the triggers.

Note: If you are generating the local binary schema cache, do not connect to the database using Trigger Location (`-trig`) and the Schema Cache File (-cache) on page 83 together.

Temp-Table Base Index (-ttbaseindex)

Use temp-table Base Index (`-ttbaseindex`) with Temp-table Index Range Size (-ttindexrangesize) on page 215 to specify the range of temp-table indexes for which you want to collect statistics.

Operating system and syntax	**UNIX / Windows**	**`-ttbaseindex` *n***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	32767	1	1	—

n

The starting temp-table index number in the range of indexes for which you want to track access statistics.

Temp-Table Base Table (-ttbasetable)

Use temp-table Base Table (`-ttbasetable`) with Temp-Table Table Range Size (-tttablerangesize) on page 215 to specify the range of temp-tables for which you want to collect statistics.

Operating system and syntax	**UNIX / Windows**	**`-ttbasetable` *n***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	32767	1	1	—

n

The starting temp-table number in the range of temp-tables for which you want to track access statistics.

Temp-table Index Range Size (-ttindexrangesize)

Use temp-table Index Range Size (`-ttindexrangesize`) to specify the number of temp-table indexes for which you want to collect statistics. The default value for `-ttindexrangesize` is 0; it specifies that index statistics are not collected.

Operating system and syntax	**UNIX / Windows**	**`-ttindexrangesize` *`n`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	0	—

`n`

The number of temp-table indexes for which you want to track access statistics.

Temp-Table Table Range Size (-tttablerangesize)

Use Temp-Table Table Range Size (`-tttablerangesize`) to specify the number of temp-tables for which you want to collect statistics. The default value for `-tttablerangesize` is 0; it specifies that statistics for temp-tables are not collected.

Operating system and syntax	**UNIX / Windows**	**`-tttablerangesize` *`n`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	0	—

`n`

The number of temp-tables for which you want to track access statistics.

Transaction End (TXE) Lock Retry Limit (-TXERetryLimit)

Use Transaction End (TXE) Lock Retry Limit (`-TXERetryLimit`) to specify the number of number of times a connection will re-try to obtain TXE lock after a 1ms nap before getting queued. The default value is 0 (no behavior change).

Operating system and syntax	UNIX / Windows	**`-TXERetryLimit`** ***`n`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Database Server	100,000	0	—	0

`n`

The number of times a connection will re-try to obtain TXE lock after a 1ms nap before getting queued. The default value is 0 (no behavior change).

The Transaction End (TXE) Lock is required for crash recovery to work properly. The lock single threads "micro-transaction" type updates within the storage engine, as well as assures that database changes are properly ordered when a transaction ends. A high degree of conflict queuing this lock can severely impact performance. The cost of managing the queue and waiting on a semaphore, as well as awakening waiting requests, while keeping things "fair" carries a high performance penalty.

When the connection spins on the TXERetryLimit, rather than waiting on a semaphore, contention shifts from the TXE Lock to the TXQ latch, but the lower level TXQ latch is a much lighter-weight mechanism, and because of that, improves the overall performance of the system. Additionally, the behavior of the TXQ latch conflicts can be further tuned using the existing parameters Spin Lock Retries (-spin) on page 200 and Nap Maximum (-napmax) on page 156.

The best value for `-TXERetryLimit` is varied, but testing by Progress Software with a value of 200 showed the best performance improvement for the scenario where multiple users are creating and deleting large numbers of records in very small transactions, which previously had displayed a severe performance penalty.

UNDO, THROW Error Handling (-undothrow)

Change the default error handling to UNDO, THROW in routine or other blocks.

Operating system and syntax	UNIX / Windows	`-undothrow` *`n`*		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

`n`

A value (either 1 or 2) that specifies whether you want to change the default error-handling to UNDO,THROW for routine-level blocks only, or for other blocks as well.

Set *`n`* to 1 to change the default behavior for routine-level blocks only. Routine-level blocks include the main block of an external procedure (.p), internal procedures, user-defined functions, methods of a class, class constructors, property accessors, and ON blocks used as database triggers with CREATE, DELETE, WRITE or ASSIGN events.

Set *`n`* to 2 to change the behavior for REPEAT blocks, FOR blocks, and DO TRANSACTION blocks in addition to all of the routine-level blocks.

Any value other than 1 or 2 will throw an error and terminate the session.

Note: The `-undothrow` startup parameter has no effect on r-code. Error handling cannot be changed after code has been compiled. The `-undothrow` parameter takes effect only during a compile of .p or .cls source files.

An alternative to the `-undothrow` startup parameter is to add either ROUTINE-LEVEL ON ERROR UNDO, THROW or BLOCK-LEVEL ON ERROR UNDO, THROW statements to your source files. For more information about these statements, see the *OpenEdge Development: ABL Reference*.

User ID (-U)

Use User ID (`-U`) together with the Password (-P) on page 172 connection parameter to specify the user ID and user account password when connecting to an OpenEdge RDBMS, either at ABL application startup or when executing the ABL `CONNECT` statement.

Operating system and syntax	UNIX / Windows	**-U *userid***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Connection	—	—	Blank ("") or OS process user ID[31]	Blank ("") or OS process user ID[31]

userid

The user ID of the user. For an OpenEdge database defining only the default blank ("") domain, you can use a non-qualified user ID (user account name only). For a database defining multiple security domains, you can use a fully qualified user ID that includes both the user account name and the name of the user's security domain separated by the domain delimiter (@). Thus, the specified *userid* can be formatted as shown in the following table, where *user-name* is a non-qualified user ID and *domain-name* is the name of the user's domain.

This format...	Writes the...
	Blank ("") user in the blank domain
@	Blank user in the blank domain
@ *domain-name*	Blank user in the specified domain
user-name	Specified user in the blank domain
user-name @	Specified user in the blank domain
user-name @ *domain-name*	Specified user in the specified domain

OpenEdge authenticates the user ID and password specified by the `-U` and `-P` parameters against the user account system defined by the authentication system configured for the user's security domain. OpenEdge identifies the authentication system to authenticate the user ID using the local database domain registry, regardless of any database options that you have set. If the user's domain is authentication-enabled and registered in the database domain registry, and a user account is found that matches the user ID and password, OpenEdge completes the database connection and sets it to the specified connection identity. Otherwise, the user is not allowed to connect to the database. If the database is multi-tenant, a successful connection also sets the user's tenancy (database tenant organization) as it is configured for the user's domain.

[31] For more information on the setting of the default user ID, see the default connection identity settings.

Note: To authenticate the user identity for a valid setting of this connection parameter, the code page of the authenticating user account system must match the setting of the startup parameter.

For backward compatibility, if you do not specify `-U` and `-P` for a database connection, OpenEdge connects the database using a *default connection identity*. OpenEdge resolves one of two possible user IDs as the default connection identity: the blank user ID or the operating system process user ID, depending on the conditions listed in the following table.

Does the database define at least one authentication-enabled domain?[35]	Are the -U and -P connection parameters specified?	Resulting user identity
NO	YES	Error: `-U` and `-P` not allowed unless enabled domains exist that support user authentication performed by OpenEdge.
NO	NO	Default connection identity: The operating system process user ID.[32], [33]
YES	NO	Default connection identity: The blank (`""`) user ID.[32], [34]
YES	YES	If the `-U`*`userid`* and `-P`*`password`* match a user account that exists for the user's authentication system, use that identity.Otherwise, OpenEdge raises an error and access is denied.

For more information on database connection identity, see the sections on connecting to a database in *OpenEdge Getting Started: Identity Management*.

The `-U` and `-P` parameters support a common user authentication model for connecting to a database from:

- The command line used to startup an ABL application
- Within a running ABL application using the `CONNECT` statement
- The command line used to startup a database utility, such as PROUTIL (using the `-userid` and `-password` parameters)

[32] Any client-principal object that you return for a database connection using the ABL GET-DB-CLIENT function, and that represents a default connection identity, cannot be used to set the identity for any other database connection or ABL session. A client-principal can only be used to set the identity for additional OpenEdge resources if it represents an authenticated user identity. For more information, see the CONNECT statement entry in *OpenEdge Development: ABL Reference*.

[33] OpenEdge sets the domain name for any default operating system user ID to "windowsid" or "unixid", depending on the operating system where the database is connected.

[34] The default blank user identity creates ambiguity in database authorization (table and field permissions) with any authenticated blank user account defined and enabled in the `_User` table accounts.

[35] A domain is *authentication-enabled* if: 1) it is configured with an authentication system that is enabled for OpenEdge-performed user authentication, 2) this authentication system has access to a source of valid user accounts, and 3) the domain is enabled for use in the database.

- The command line or other configuration used to startup an OpenEdge SQL (JDBC or ODBC) application

The authentication model for these connections is supported by a common security domain configuration within the OpenEdge database that you define using OpenEdge Database Administration. The requirement for all users connecting to a database, whether from an ABL application, an SQL application, or a database utility, is that their domain must be authentication-enabled (supports OpenEdge-performed user authentication). A domain enabled **only** for single sign-on (SSO) or for application-defined user authentication cannot authenticate a user who is connecting to a database from an ABL application, SQL application, or database utility.

In addition, if a domain is configured with an ABL authentication plugin, users cannot authenticate in this domain from an SQL application or from the startup command line of an ABL application. To connect a database with a user identity that authenticates using an ABL plugin, you must use the ABL `CONNECT` statement within the ABL session, or connect the database from the command line using a different identity, then use the ABL `SET-DB-CLIENT` function or `SET-CLIENT( )` method on the `SECURITY-POLICY` system handle to authenticate and set the connection identity using the ABL plugin.

For more information on the `CONNECT` statement, see the statement entry in *OpenEdge Development: ABL Reference*. Note that, except for use of an ABL plugin or when otherwise noted, the features described for connecting a database using the `CONNECT` statement also apply to connecting a database on the startup command line for an ABL application.

For more information on using security domains and authentication systems in OpenEdge, see *OpenEdge Getting Started: Identity Management*. For more information on using ABL authentication plugins, see the sections on application security and authentication in *OpenEdge Development: Programming Interfaces*.

For more information on the features for authenticating the user of an OpenEdge SQL application connection to an OpenEdge database, see the chapters on using JDBC and ODBC clients in *OpenEdge Data Management: SQL Development*.

For more information on how to specify a user ID and password when connecting from a database utility, see the reference entry for the utility in *OpenEdge Data Management: Database Administration*.

Note: With certain DataServers, the `-U` and `-P` parameters also pass DataServer login information to the foreign (non-OpenEdge) database. For more information, see your DataServer documentation.

Use OS Locale (-useOsLocale)

Use the Use OS Locale (`-useOsLocale`) parameter to set certain default localization options for OpenEdge clients.

Operating system and syntax	Windows	**`-useOsLocale`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

This parameter causes the AVM to set certain `SESSION` system handle attributes based on the current Windows locale settings.

The following table lists these `SESSION` system handle attributes and shows the startup parameters that individually affect the value of each attribute.

SESSION system handle attribute	Associated startup parameters
`DATE-FORMAT`	Date Format (-d) on page 101
`NUMERIC-DECIMAL-POINT`	Fractional Separator (-numdec) on page 168
`NUMERIC-FORMAT`	European Numeric Format (-E) on page 112, Fractional Separator (-numdec) on page 168, Thousands Separator (-numsep) on page 169
`NUMERIC-SEPARATOR`	Thousands Separator (-numsep) on page 169

If any of the startup parameters shown in this table appear on the same command line

with the `-useOsLocale` parameter, any that follow `-useOsLocale` override the setting

from `-useOsLocale` and any that `-useOsLocale` follows are overridden by the setting from `-useOsLocale`.

For more information on these `SESSION` system handle attributes, see *OpenEdge Development: ABL Reference*.

In a session accessing .NET objects, Progress Software Corporation recommends that you use this startup parameter to resolve potential discrepancies between localization settings in your ABL session and the embedded .NET Common Language Runtime (CLR). Using this parameter makes your ABL session conform to Windows (and therefore .NET) settings. If you want to override these settings in your ABL session, you can also propagate these overrides to the CLR by instantiating a .NET System.Globalization.Cultureinfo class and making the appropriate settings using the properties and methods of this instance.

Use Widget ID (-usewidgetid)

Use the Use Widget ID (`-usewidgetid`) startup parameter to enable application-defined widget IDs for ABL widgets defined in your OpenEdge GUI application.

Operating system and syntax	Windows	**`-usewidgetid`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

Specifying application-defined widget IDs for ABL widgets defined in your OpenEdge GUI application allows you to identify those widgets at runtime while testing your application with a third-party automated test tool. When specified, the AVM uses any application-defined widget IDs when creating the widgets at runtime, instead of using the widget IDs it normally generates by default. The widget ID value of any given widget remains the same across OpenEdge sessions, unless you purposely change the value using the `WIDGET-ID` attribute for that widget. This allows a third-party automated test tool to identify the same widget consistently each time you run the tool with your application.

For more information about specifying application-defined widget IDs using the `WIDGET-ID` attribute with ABL widgets, see *OpenEdge Development: ABL Reference*. For more information about using application-defined widget IDs when testing OpenEdge GUI applications with third-party automated test tools, see *OpenEdge Development: Programming Interfaces*.

Version 6 Colon (-v6colon)

Use Version 6 Colon (`-v6colon`) to direct the AVM to perform colon alignment for unlabeled fields, as in Version 6.

Operating system and syntax	**UNIX / Windows**	**`-v6colon`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

If you do not specify this parameter, the AVM moves the field two characters to the right.

Version 6 Query (-v6q)

Use Version 6 Query (`-v6q`) to direct the OpenEdge server to use only a single index when resolving `FOR EACH` statements.

Operating system and syntax	**UNIX / Windows**	**`-v6q`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

If you do not use this parameter, the server might use more than one index. Using more than one index is more efficient, but might change the order in which records are returned.

The `-v6q` parameter also causes cursor repositioning with `FOR EACH` statements, as in Version 6.

Windows Passthrough Printing (-Wa -wpp)

Use Windows Passthrough Printing (`-Wa -wpp`) to enable WPP mode for the session.

Operating system and syntax	Windows	**`-Wa -wpp`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

You can only use this startup parameter at the command line. It is ignored when used in a `.pf` file. It must be the last parameter in the command.

Windows Single Session (-wss)

Use Windows Single Session (`-wss`) to limit a user to running only one OpenEdge client session at a time.

Operating system and syntax	Windows	**`-wss`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

This parameter applies only to a Windows GUI client. If an existing OpenEdge session is running on the machine when `-wss` is specified, the new session terminates and the windows of the existing session are brought forward to alert the user. To avoid unnecessary processing of other parameters, use the `-wss` parameter first on the command line.

If more than one OpenEdge application is installed on a user's machine, do not use the `-wss` parameter else it will prevent a user from running two or more applications at the same time. In addition, the `-wss` parameter does not recognize versions of OpenEdge prior to Version 9.0.

Windows Exit - No Dialog (-wy)

Use Windows Exit - No dialog (`-wy`) to prevent OpenEdge from prompting you for confirmation when Windows is shut down with OpenEdge active.

Operating system and syntax	**Windows**	**`-wy`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

Statistics (-y)

Use Statistics (`-y`) to collect procedure access and usage statistics throughout the OpenEdge session.

Operating system and syntax	**UNIX / Windows**	**`-y`**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

At session end, the AVM writes these statistics to the default output file `client.mon`. For detailed information on the output written to the `client.mon` file, see *OpenEdge Deployment: Managing ABL Applications*.

At startup, the `-y` parameter sends a report of all the startup parameters to the output file. This report includes all default values, overridden values, and values you set at startup. Unlike the other statistics that the `-y` parameter collects, this report is not written at session end and is not affected by the `SHOW-STATS` statement.

Note: The specific statistics displayed might change periodically as new OpenEdge features are implemented.

The edit buffer map statistics are written each time a user exits from OpenEdge or uses the `SHOW-STATS` statement. These lines list the procedures currently in the edit (or execution) buffer and their r-code sizes.

Note: If you cannot execute `SHOW-STATS` from the Procedure Editor or cannot add the statement to your ABL code (for example, if while using Run-time OpenEdge), you can specify Statistics with CTRL+C (-yc) on page 225 instead of the `-y` parameter. Both parameters behave the same way, except `-yc` lets you use **CTRL+C** as a substitute for the `SHOW-STATS` statement.

The program-access statistics are written to the output file when the session ends or when you use the `SHOW-STATS` statement. In the output, temp file reads and writes are reads and writes to the `SRT` file, which stores each user's session compiles and active r-code files. The **Bytes** column is a cumulative total. The Stat file checks are recorded because they represent a relatively time-consuming system call. Unless you invoke Quick Request (-q) on page 185, the AVM makes a stat call each time a precompiled subprocedure is called with the `RUN` statement.

The AVM places the default output file (`client.mon`) in the current working directory; however, you can specify a different output file by using the `CLIENTMON` environment variable. Simply set `CLIENTMON` to point to the file you want to use. For example, in a UNIX environment, if you wanted to use a file named `stats` in the `/usr/tmp` directory, enter the following command at the system prompt:

```
CLIENTMON=/usr/tmp/stats; export CLIENTMON
```

Statistics with CTRL+C (-yc)

Use Statistics with CTRL+C (`-yc`) to allow you to press **CTRL+C** as a substitute for executing the `SHOW-STATS` statement.

Operating system and syntax	UNIX / Windows	**-yc**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

Use this parameter if you cannot execute `SHOW-STATS` from the Procedure Editor or if you cannot add the statement to your ABL code (for example, if while using run-time OpenEdge).

When you specify this parameter, it disables **CTRL+C** as an interrupt signal. In addition, pressing **CTRL+C** only emulates `SHOW-STATS` without the `CLEAR` option.

Segment Statistics (-yd)

Use Segment Statistics (`-yd`) to write segment statistics to the client monitor file (`client.mon`, by default).

Operating system and syntax	**UNIX / Windows**	**-yd**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

Segment statistics provide:

- A breakdown of an r-code file by segment type, including the number of segments and their total size
- Read and write access to the sort file by segment type and access type, including the number of times the sort file was accessed and the number of bytes read from/written to the sort file

Four Digit Year Default (-yr4def)

Use Four Digit Year Default (`-yr4def`) to output a four-digit year from the `EXPORT`, `MESSAGE` and `PUT UNFORMATTED` statements that may use a two-digit year.

Operating system and syntax	**UNIX / Windows**	**-yr4def**		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

This option causes all dates to use a four-digit year, regardless of what is specified by the Century year Offset (`-yy`) startup parameter.

If you do not use this parameter, the `EXPORT`, `MESSAGE` and `PUT UNFORMATTED` statements produce a two-digit year for the date if it is within the 100 year period defined by the Century Year Offset (-yy) on page 228 startup parameter. If the date is outside of that 100 year period, these statements produce a four-digit year for the date.

Using `-yr4def` does not affect statements such as `DISPLAY`, `UPDATE` and `PUT` (without `UNFORMATTED`), nor any default schema or program variable date formats, which continue to use the default format 99/99/99. These statements have the `FORMAT` phrase where an overriding date format can be given, if the 4-digit year is desired, or the fields and variables can be defined with the explicit 99/99/9999.

Statistics With Cross-reference (-yx)

Use Statistics with Cross-reference (-yx) to collect procedure call statistics and write them to an output file.

Operating system and syntax	UNIX / Windows	-yx		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session	—	—	—	—

With this parameter, you can monitor information to answer questions:

- How many calls were made in a given period of time?
- How long did a procedure spend executing?
- How often was a procedure swapped to and from the temporary file (SRT file)?

The AVM places the default output file for the -yx parameter, proc.mon, in your current working directory. However, you can specify a different output file by using the PROCMON environment variable. Simply set PROCMON to point to the file you want to use.

For example, in a UNIX environment, if you want to use a file named stats in the /usr/tmp directory, enter the following on the command line:

```
PROCMON=/usr/tmp/stats; export PROCMON
```

Use the SHOW-STATS statement to write the procedure call statistics to the output file. The CLEAR option sets all counters and timers (such as Calls and Time) to 0. When you exit OpenEdge, the -yx parameter writes the procedure call statistics to the output file whether or not you use SHOW-STATS.

The following example shows sample procedure call statistics as they appear in the output file:

```
                Wed Feb 6 16:19:05 2002

  Procedure call statistics:      16:19:21

   Caller      Callee      Load Size    Calls   Rd Bytes     Reread   Time
   <top>       _edit         3789         1         3789         0      16
   _edit       _adeload      9542         1         9542         0      10
   _edit       _prostar      6931         1         6931         0      23
   _prostar    _login        5087         1         5087         0      73
   _edit       _setcurs       562         2          562         0      20
   _edit       _proedit    181029         1       181029         0   12180
   _proedit    _toollic      2570         1         2570         0     268
   _proedit    _tmpfile      1510         2         1510         0     140
   _proedit    _kvlist       2624         2         2624         0      26
   _proedit    _setcurs       562         4           0          0      63
   _proedit    _adehelp      3279         1         3279         0      97
```

The fields that appear in the output file `proc.mon` are as follows:

- **Caller** — The names of any calling procedures. The word `<top>` indicates there was no calling procedure, and you ran the called procedure directly from the Procedure Editor.
- **Callee** — The names of the called procedures.
- **Load Size** — The size, in bytes, of each called procedure as the AVM loads it into memory. If you see a load size of 0, the called procedure is an uncompiled source procedure (`.p` file), or an r-code file (`.r` file) that was previously loaded into memory.
- **Calls** — The number of times the **Caller** procedure calls the **Callee** procedure.
- **Rd Bytes** — Generally, the called procedure's load size. However, if the procedure is swapped out of memory and later restored, **Rd bytes** equals the procedure's Load Size added to the number of bytes read from the `SRT` file. **Rd Bytes** grows larger each time the procedure is swapped out of memory and restored from the `SRT` file.
- **Reread** — The number of bytes the AVM reads from the `SRT` file to restore a Caller procedure that was overwritten. The AVM restores only what is necessary to continue executing the Caller procedure. In some cases, this is less than the amount swapped to the `SRT` file.
- **Time** — The total execution time of the called procedure, in milliseconds.

Century Year Offset (-yy)

Use Century Year Offset (`-yy`) to determine the start of the 100-year period in which a date with a two-digit year is valid.

Operating system and syntax	**UNIX / Windows**	**`-yy` *`n`***		
Use with	**Maximum value**	**Minimum value**	**Single-user default**	**Multi-user default**
Client Session, Database Server	—	—	1950	1950

`n`

A four-digit year (1990, for example). The default is 1950.

Some OpenEdge applications reserve only two digits for the year in the date format. When, for example, `-yy` is set at 1950, the AVM determines if the two-digit year value is greater or less than 50. If the year is greater than 50, the AVM assumes that the date is in the twentieth century. If the year is less than 50, the AVM assumes that the date is in the twenty-first century.

The following table shows some examples of `-yy`.

-yy	Year (as supplied in DATE format)	Result of year function
1900	50-99	1950-1999
	00-49	1900-1949

-yy	Year (as supplied in DATE format)	Result of year function
1950	50-99	1950-1999
	00-49	2000-2049
1980	80-99	1980-1999
	00-79	2000-2079

Notice that all two-digit year values expand into the 100-year period beginning with `-yy`.

To test the effect of `-yy`, start OpenEdge with a different `-yy` value and run the following procedure:

```
DEFINE VARIABLE dDate AS DATE NO-UNDO.
DISPLAY "Enter date or Press F4 to end." WITH FRAME msg.
REPEAT:
  SET dDate LABEL "Date" WITH SIDE-LABELS.
  DISPLAY YEAR(dDate) FORMAT "9999" LABEL "=> Year" WITH SIDE-LABELS.
END.
```

If you use a hard-coated date containing a two-digit year in a `.p` file, the AVM honors the `-yy` parameter and expands the two-digit year to a four-digit year during compilation. However, this may not match the runtime `-yy`. For this reason, Progress Software Corporation recommends that you use four digit years for hard-coated dates in programs. For example:

```
DEFINE VARIABLE hdate AS DATE INITIAL 01/01/2001.
```

Note: When you dump or load any database, the `-yy` setting you load with must match the `-yy` setting that was used for dumping, unless you use the Four Digit Year Default (-yr4def) on page 226 startup parameter for dumping.

This startup parameter provides the same functionality as the `SESSION:YEAR-OFFSET` attribute.

Index