Progress®



OpenEdge® Management and OpenEdge Explorer:
Getting Started with Multi-tenancy

OpenEdge®

Copyright

© **2019 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.**

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

Corticon, DataDirect (and design), DataDirect Cloud, DataDirect Connect, DataDirect Connect64, DataDirect XML Converters, DataDirect XQuery, DataRPM, Deliver More Than Expected, Icenium, Kendo UI, Kinvey, NativeScript, OpenEdge, Powered by Progress, Progress, Progress Software Developers Network, Rollbase, SequeLink, Sitefinity (and Design), Sitefinity, 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. Analytics360, AppServer, BusinessEdge, DataDirect Autonomous REST Connector, DataDirect Spy, SupportLink, DevCraft, Fiddler, JustAssembly, JustDecompile, JustMock, Kinvey Chat, NativeChat, NativeScript Sidekick, OpenAccess, ProDataSet, Progress Results, Progress Software, ProVision, PSE Pro, SmartBrowser, SmartComponent, SmartDataBrowser, SmartDataObjects, SmartDataView, SmartDialog, SmartFolder, SmartFrame, SmartObjects, SmartPanel, SmartQuery, SmartViewer, SmartWindow, and WebClient are trademarks or service marks of Progress Software Corporation and/or its subsidiaries or affiliates in the U.S. and other countries. Java is a registered trademark of Oracle and/or its affiliates. Any other marks contained herein may be trademarks of their respective owners.

February 2019

Last updated with new content: Release 12.0.0

Updated: 2019/03/01

Table of Contents

Preface....7
Purpose....7
Audience....7
Organization....8
Using ABL documentation....8
References to ABL compiler and run-time features....8
References to ABL data types....8
Typographical conventions....9
Examples of syntax descriptions....10
Long syntax descriptions split across lines....11
Complex syntax descriptions with both required and optional elements....12
OpenEdge messages....12
Obtaining more information about OpenEdge messages....13

Chapter 1: Getting Started with Multi-tenancy in the Database Administration Console....15
Overview....16
Step 1: Starting the Database Administration Console in OpenEdge Management or OpenEdge Explorer....16
Step 2: Accessing the Database Administration Console....18
Step 3: Adding a database connection....19
Step 4: Enabling the database for multi-tenancy....19
Step 5: Creating a super-tenant....20
Step 6: Creating a domain for the super-tenant....20
Step 7: Adding a user for the super-tenant....20
Step 8: Setting up the super-tenant's database login user credentials....21
Step 9: Creating a regular tenant, its domain, and its users....21
Step 10: Enabling tables for multi-tenancy....21
For more information about multi-tenancy in OpenEdge....22

Preface

For details, see the following topics:

- Purpose
- Audience
- Organization
- Using ABL documentation
- Typographical conventions
- Examples of syntax descriptions
- OpenEdge messages

Purpose

This manual describes how to get started with database multi-tenancy.

Audience

This manual is intended for OpenEdge® Management and OpenEdge Explorer users as well as OpenEdge Management database and system administrators.

Organization

Getting Started with Multi-tenancy in the Database Administration Console on page 15

Describes how to get started in the console by setting up a multi-tenant database and working with the database and its tenants, users, and contents.

Using ABL documentation

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 go to https://docs.progress.com.

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 documentation 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.
`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.

Convention	Description
|	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

Getting Started with Multi-tenancy in the Database Administration Console

OpenEdge Management/OpenEdge Explorer now includes the Database Administration Console, the configuration tool that allows you to manage and work with databases enabled for multi-tenancy. Integrated into the OpenEdge Management/OpenEdge Explorer management console, the Database Administration Console allows you to perform a variety of different tasks related to multi-tenancy, such as configure connections for local and remote databases; work with database tenants, sequences, users, domains, and groups; update database schema, and create and work with tenant templates.

This guide is intended to get you started with setting up a multi-tenant database and lead you to the point where you can work with the database and its tenants, users, and contents. The guide is designed to be used in conjunction with *OpenEdge Management and OpenEdge Explorer: Configuring Multi-tenancy*.

For details, see the following topics:

- Overview
- Step 1: Starting the Database Administration Console in OpenEdge Management or OpenEdge Explorer
- Step 2: Accessing the Database Administration Console
- Step 3: Adding a database connection
- Step 4: Enabling the database for multi-tenancy
- Step 5: Creating a super-tenant
- Step 6: Creating a domain for the super-tenant
- Step 7: Adding a user for the super-tenant
- Step 8: Setting up the super-tenant's database login user credentials

- Step 9: Creating a regular tenant, its domain, and its users
- Step 10: Enabling tables for multi-tenancy
- For more information about multi-tenancy in OpenEdge

Overview

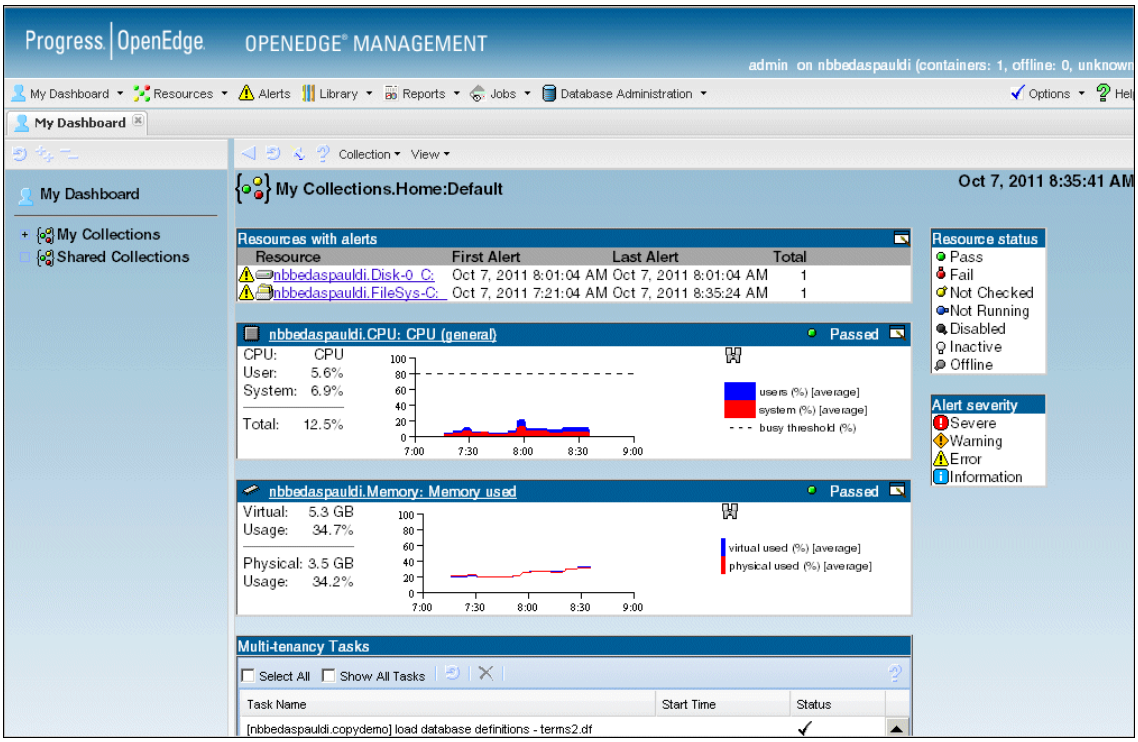
The focus of this guide is on getting you up and running with a database enabled for multi-tenancy. The guide begins at the point at which you've completed your OpenEdge installation and are logging into the Database Administration Console, which is a component of the larger OpenEdge Management or OpenEdge Explorer management console. As you follow the steps provided, you will be guided through setting up a database connection; setting up a super-tenant, its domain, and its users; establishing secure login credentials for the super-tenant; and converting existing tables so that they are enabled for multi-tenancy and/or creating new tables and then enabling them.

This guide does not tell you **how** to perform the various tasks related to multi-tenancy. Instead, this guide provides you with a roadmap and recommendations for starting to work with multi-tenancy. Included are the tasks you must perform to configure a database for multi-tenancy and then be able to work with and manage tenants and data in the database.

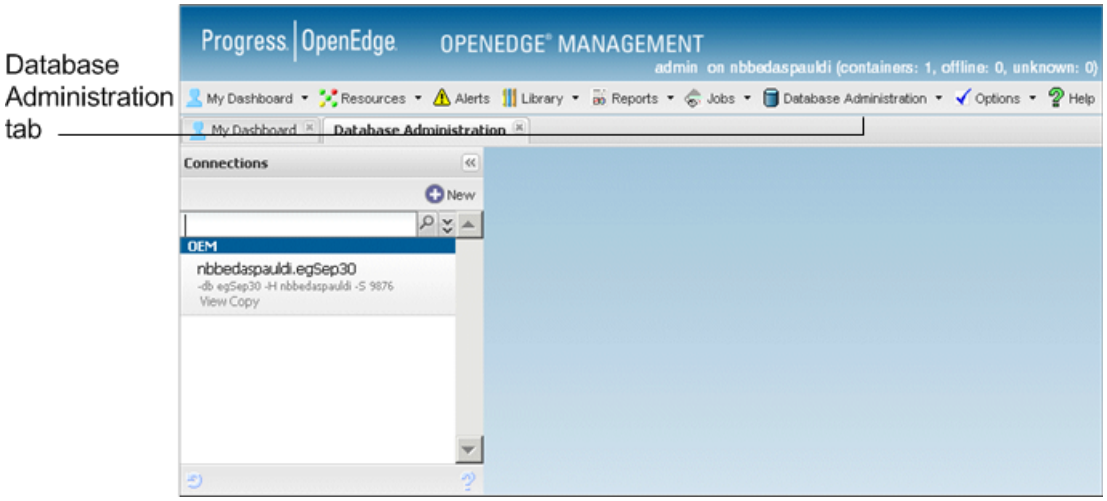
Step 1: Starting the Database Administration Console in OpenEdge Management or OpenEdge Explorer

Once you complete the OpenEdge installation, you can start the Database Administration Console in OpenEdge Management or OpenEdge Explorer. Note that this guide refers to two consoles:

- The OpenEdge Management and OpenEdge Explorer management console, which comprises functionality for following areas, as indicated by the tabs in the menu bar: **My Dashboard**, **Resources**, **Alerts**, **Library**, **Reports**, **Jobs**, **Database Administration**, **Options**, and **Help**:



- The Database Administration Console, accessible from within the management console by the **Database Administration** tab:



The management console and the Database Administration Console look the same in OpenEdge Explorer as they do in OpenEdge Management. The only difference you find when using OpenEdge Explorer is that any OpenEdge Management-only functionality is disabled in the management console when you do not have the required OpenEdge Management license.

OpenEdge Management/OpenEdge Explorer runs in a Web browser, making it accessible in Windows and on supported UNIX platforms. It is recommended that you use a recent version of a common Web browser to take full advantage of the functionality.

To start the Database Administration Console in OpenEdge Management or OpenEdge Explorer in a browser and then log in:

1. Choose one:

- Open a Web browser, and enter the URL `http://host:port` in the address or location field. The host is the name of the machine where OpenEdge Management/OpenEdge Explorer is installed; the default port is 9090.
- From the Windows Desktop on your local host, select **Start** > **Programs (or All Programs)** > **Progress** > **OpenEdge** > **Database Administration Console**> to start OpenEdge Management/OpenEdge Explorer.

  If you have only OpenEdge Explorer installed, select **Start** > **Programs (or All Programs)** > **Progress** > **OpenEdge** > **OpenEdge Explorer**.

The logon window appears.

2. Provide the user name and password. The first time you log on to OpenEdge Management or OpenEdge Explorer, you must use the default user name, which is **admin**, and the default password, which is **admin**.
3. Do the following:

   a) Complete the **OpenEdge Management Configuration** page to make your initial configuration choices.

   This includes resetting your login password, choosing whether to have OpenEdge Management or OpenEdge Explorer start automatically, choosing a location for the OpenEdge Management Trend Database, specifying the Web server port, choosing the e-mail server and default email recipient, and establishing SNMP Adapter settings.

   Click **Submit**.

For details, see:	Choosing initial configuration options in *OpenEdge Management and OpenEdge Explorer: Getting Started*

   b) Complete the **Getting Started** page (OpenEdge Management only).

   This includes adding one or more external databases and defining monitoring and trending options for file systems and disks.

   Click **Submit**.

For details, see:	Setting up the Getting Started page for OpenEdge Management in *OpenEdge Management and OpenEdge Explorer: Getting Started*

4. Click **OK** to restart the Web server, and then log in with your user name and new password.

Step 2: Accessing the Database Administration Console

Once you log in, you can begin working with multi-tenancy in the Database Administration Console.

From the management console menu bar, click the **Database Administration** tab. A **Connections** view opens in the left pane.

Step 3: Adding a database connection

To work with a database in the Database Administration Console, you must have a connection established for it.

You can configure any of the following database connections:

- A managed connection, which refers to a database that is either of the following:
  - **Managed database** — A database that is being managed by OpenEdge Management or OpenEdge Explorer. The database can reside on the local host or on a remote AdminServer. For a managed database, the connection is made through shared memory.
  - **Scripted database** — A database that you start with a script outside of the OpenEdge Management or OpenEdge Explorer environment. The database can reside on the local host or on a remote AdminServer.
- An unmanaged database connection, which refers to a database that:
  - Is not being managed by OpenEdge Management or OpenEdge Explorer.
  - You are using only to perform database administration tasks related to multi-tenancy.

  This type of database connection is new in this release of OpenEdge.

Decide if the connection you are setting up will be to a database on a local machine or a remote machine. If you want to set up a connection to a database on a remote machine, you must **first** set up remote configuration.

You can then set up the local or remote database connection.

For details about , setting up remote configuration **before** setting up the database connection, see:	**1.** Reviewing the remote configuration requirements in *OpenEdge Management and Explorer: Configuring Multi-tenancy* **2.** Ungluing OpenEdge Management/OpenEdge Explorer from OpenEdge in *OpenEdge Management and Explorer: Configuring Multi-tenancy* **3.** Setting up the local host machine in *OpenEdge Management and Explorer: Configuring Multi-tenancy* **4.** Setting up the remote AdminServer in *OpenEdge Management and Explorer: Configuring Multi-tenancy*

For details about setting up a database connection, see:	Setting up the connection in *OpenEdge Management and Explorer: Configuring Multi-tenancy*User credentials considerations in *OpenEdge Management and Explorer: Configuring Multi-tenancy*

Step 4: Enabling the database for multi-tenancy

You can enable a database for multi-tenancy in either of the following ways:

- By using the following command from the folder where the database resides:

```
proutil <database-name> -C enableMultitenancy
```

- From the Database Administration Console.

For details, see:	Enabling a database for multi-tenancy in *OpenEdge Management and Explorer: Configuring Multi-tenancy*

Step 5: Creating a super-tenant

Once you enable a database for multi-tenancy, create a super-tenant.

There are three varieties of tenants:

- **Super-tenant** — A tenant that does not have its own data but can access all shared tables and all multi-tenant data in the database.
- **Regular tenant** — A tenant that can access its own data, data for any other tenants in the same tenant group, or data in shared tables. (A shared table is one whose data is available to all tenants in the database.)
- **Default tenant** — A tenant that does not establish tenancy as part of authentication to the database. Because the user has not established tenancy, the default tenant can access only shared table data within the database or the default partition of a multi-tenant table. There is only one default tenant per multi-tenant database.

It is recommended that you set up at least one super-tenant to manage the database, and it is likely that you will want to set up at least one regular tenant as well.

For details, see:	Creating a new tenant for a multi-tenant database in *OpenEdge Management and Explorer: Configuring Multi-tenancy*

Step 6: Creating a domain for the super-tenant

For each tenant, you must create a domain. There are several predefined domains available for the database, or you can create your own. You can have multiple domains per tenant.

Start by creating a domain for the super-tenant.

For details, see:	Adding a tenant domain in *OpenEdge Management and Explorer: Configuring Multi-tenancy*

Step 7: Adding a user for the super-tenant

Once you have added a domain, you must create a user who can log in to the super-tenant in the database. You can have multiple users per domain.

For details, see:	Adding a user to a tenant in *OpenEdge Management and Explorer: Configuring Multi-tenancy*

Step 8: Setting up the super-tenant's database login user credentials

By default, connections for databases defined in OpenEdge Management/OpenEdge Explorer use the blank user ID. In order to access tenant data, you must set up the database connection to use the credentials for either a user for a regular tenant or for a super-tenant. Because super-tenants can access all users' data, establishing a database connection as a super-tenant will allow you to perform operations such as dumping and loading of data and editing sequences for any tenant. If you establish a connection with the credentials of a user for a regular tenant, you can perform operations for only that tenant.

For details, see:	Setting up a database connection login in *OpenEdge Management and Explorer: Configuring Multi-tenancy*

Step 9: Creating a regular tenant, its domain, and its users

Once you create and establish credentials for logging in to a super-tenant, create a regular tenant, its domain, and one or more of its users.

For details, see:	**1.** Creating a new tenant or tenant program *OpenEdge Management and Explorer: Configuring Multi-tenancy* **2.** Adding a tenant domain *OpenEdge Management and Explorer: Configuring Multi-tenancy* **3.** Adding a user to a tenant *OpenEdge Management and Explorer: Configuring Multi-tenancy*

Step 10: Enabling tables for multi-tenancy

From the management console, you can choose one or more tables not enabled for multi-tenancy, and enable them. You can then make decisions about partition allocation for any of the table's fields and indexes that have not yet been allocated.

Note that when you enable a table for multi-tenancy, you are affecting all tenants for a selected database.

For details, see:	Enabling a table for multi-tenancy *OpenEdge Management and Explorer: Configuring Multi-tenancy*

For more information about multi-tenancy in OpenEdge

In addition to the getting started information provided in this guide, there are other sources of information available to you about multi-tenancy.

OpenEdge Management and OpenEdge Explorer: Configuring Multi-tenancy — Describes in detail how to configure database connections, manage multi-tenancy-enabled databases, work with tenants, upload tenant schema changes, manage users and domains, work with tenant templates and groups, and manage sequences and sequence values.

OpenEdge Getting Started: Multi-tenancy Overview — Describes multi-tenancy, provides details about its features, and includes a glossary of terms.

In addition to these documents, several videos related to multi-tenancy are available in this location on Progress Communities:

https://community.progress.com/technicalusers/w/openedgegeneral/1117.openedge-educational-videos.aspx