OpenEdge® Getting Started: Migrating to OpenEdge 11.7
Copyright

© 2017 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.

Business Making Progress, Corticon, Direct (and design), DataDirect Cloud, DataDirect Connect, DataDirect Connect64, DataDirect XML Converters, DataDirect XQuery, Deliver More Than Expected, Icenium, Kendo UI, Making Software Work Together, NativeScript, OpenEdge, Powered by Progress, Progress, Progress Software Developers Network, Rollbase, 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, Analytics360, AppsAlive, AppServer, Arcade, BravePoint, BusinessEdge, DataDirect Spy, DataDirect SupportLink, DevCraft, DigitalFactory, Fiddler, Future Proof, High Performance Integration, JustCode, JustDecompile, JustMock, JustTrace, OpenAccess, ProDataSet, Progress Arcade, Progress Profiles, Progress Results, Progress RFID, 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 affiliates 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:

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

March 2017

Last updated with new content: Release 11.7

Updated: 2017/03/24
# Table of Contents

**Preface**............................................................................................................7  
- Purpose........................................................................................................7  
- Audience.....................................................................................................8  
- Organization.............................................................................................8  
- Using ABL documentation.........................................................................8  
  - References to ABL compiler and run-time features..........................8  
  - References to ABL data types...............................................................9  
- Typographical conventions........................................................................9  
- Examples of syntax descriptions.............................................................10  
  - Long syntax descriptions split across lines......................................12  
  - Complex syntax descriptions with both required and optional elements..12  
- Example procedures..................................................................................12  
- OpenEdge messages..................................................................................13  
  - Obtaining more information about OpenEdge messages.................14

**Chapter 1: Progress Application Server for OpenEdge 11.7 Migration Tasks**.................................................................................................15  
- Migrating existing PAS for OpenEdge instances.....................................15  
- SSL Support Deprecated..........................................................................16

**Chapter 2: Progress Developer Studio for OpenEdge Migration Tasks**..................................................................................................17  
- Migrating an ABL Web App Project.........................................................17

**Chapter 3: Migrating OpenEdge Business Process Management**.........19  
- Savvion 7.6.X/8.0 to Progress OpenEdge 11.3.0 Migration Process........19  
- Before you begin.....................................................................................20  
- Points to remember................................................................................21  
- Migrating to OpenEdge 11.3.0.................................................................23  
- Migrating from OpenEdge 11.6.x and earlier releases.........................41  
- Before you begin.....................................................................................42  
- UNIX platform.........................................................................................42  
- Configuring the OpenEdge database.....................................................43  
- Using the Data Migration Utility............................................................43  
- Decision step migration.........................................................................45
Preface

For details, see the following topics:

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

Purpose

This migration guide contains information on the manual configuration tasks that you may need to perform after completing an OpenEdge release 11.7 installation. Manual configuration is required if you are upgrading to OpenEdge 11.7 and already have workspaces, instances, projects, etc. from the following products:

- Progress Application Server for OpenEdge (11.6.x and earlier releases)
- Progress Developer Studio for OpenEdge (11.6.x and earlier releases)
- Savvion (7.6.x/8.0)
- OpenEdge Business Process Management (11.6.x and earlier releases)
Audience

This manual is intended for any OpenEdge administrator or developer who is responsible for installing OpenEdge 11.7 products.

Organization

Progress Application Server for OpenEdge 11.7 Migration Tasks on page 15
Describes how to update existing PAS for OpenEdge instances (i.e. those created on 11.6.x and prior releases) so they can start and run on an OpenEdge 11.7 platform.

Progress Developer Studio for OpenEdge Migration Tasks on page 17
Describes how to update existing PAS for OpenEdge projects and web applications so you can continue application development after an OpenEdge 11.7 installation.

Migrating OpenEdge Business Process Management on page 19
Describes how to migrate existing Savvion sources to OpenEdge 11.3, and existing OpenEdge Business processes to OpenEdge 11.7 so you can make use of a new set of functionalities of Progress Developer Studio for OpenEdge.

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 see the OpenEdge Product Documentation Overview page on Progress Communities:

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:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bold</strong></td>
<td>Bold typeface indicates commands or characters the user types, provides emphasis, or the names of user interface elements.</td>
</tr>
<tr>
<td><em>Italic</em></td>
<td>Italic typeface indicates the title of a document, or signifies new terms.</td>
</tr>
<tr>
<td>SMALL, BOLD CAPITAL LETTERS</td>
<td>Small, bold capital letters indicate OpenEdge key functions and generic keyboard keys; for example, GET and CTRL.</td>
</tr>
<tr>
<td>KEY1+KEY2</td>
<td>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.</td>
</tr>
<tr>
<td>KEY1 KEY2</td>
<td>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.</td>
</tr>
</tbody>
</table>

Syntax:

<table>
<thead>
<tr>
<th>Fixed width</th>
<th>A fixed-width font is used in syntax, code examples, system output, and file names.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Fixed-width italics</strong></td>
<td>Fixed-width italics indicate variables in syntax.</td>
</tr>
<tr>
<td><strong>Fixed-width bold</strong></td>
<td>Fixed-width bold italic indicates variables in syntax with special emphasis.</td>
</tr>
</tbody>
</table>
**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 . . . ] }
```

Example procedures

OpenEdge documentation may provide example code that illustrates syntax and concepts. You can access many of the example files, and details for installing them, from the following locations:

- A self-extracting Documentation and Samples file available on the OpenEdge download page of the Progress Software Download Center
Once installed, you can locate the example files in the following paths under the OpenEdge Documentation and Samples installation directory:

<table>
<thead>
<tr>
<th>This directory . . .</th>
<th>Contains examples for the following documents . . .</th>
</tr>
</thead>
<tbody>
<tr>
<td>src\prodoc\dotnetobjects</td>
<td>OpenEdge Development: GUI for .NET Programming</td>
</tr>
<tr>
<td>src\prodoc\dynamics</td>
<td>The Progress Dynamics documentation</td>
</tr>
<tr>
<td>src\prodoc\getstartoop</td>
<td>OpenEdge Development: Object-oriented Programming</td>
</tr>
<tr>
<td>src\prodoc\handbook</td>
<td>OpenEdge Getting Started: ABL Essentials</td>
</tr>
<tr>
<td>src\prodoc\interfaces</td>
<td>OpenEdge Development: Programming Interfaces</td>
</tr>
<tr>
<td>src\prodoc\json</td>
<td>OpenEdge Development: Working with JSON</td>
</tr>
<tr>
<td>src\prodoc\langref</td>
<td>OpenEdge Development: ABL Reference</td>
</tr>
<tr>
<td>src\prodoc\prodatasets</td>
<td>OpenEdge Development: ProDataSets</td>
</tr>
<tr>
<td>src\prodoc\tranman</td>
<td>OpenEdge Development: Translation Manager</td>
</tr>
<tr>
<td>src\prodoc\visualdesigner</td>
<td>OpenEdge Getting Started: Introducing Progress Developer Studio for OpenEdge Visual Designer</td>
</tr>
<tr>
<td>src\prodoc\xml</td>
<td>OpenEdge Development: Working with XML</td>
</tr>
<tr>
<td>src\samples\open4gl\java</td>
<td>OpenEdge Development: Java Open Client</td>
</tr>
</tbody>
</table>

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.
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:

   ```bash
   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**.
Progress Application Server for OpenEdge
11.7 Migration Tasks

The 11.7 Progress Application Server for OpenEdge release includes upgrades to 3rd party products including Java (to version 8), Apache Tomcat (to version 8.5.x), and the Spring Security framework. These products have backward compatibility issues. Where possible, these issues are resolved during the OpenEdge 11.7 installation. However some issues (for example, when key functionality is no longer supported) require configuration updates after the 11.7 install.

For details, see the following topics:

• Migrating existing PAS for OpenEdge instances
• SSL Support Deprecated

Migrating existing PAS for OpenEdge instances

The core Apache Tomcat server was upgraded to Version 8.5.x in the 11.7 release of PAS for OpenEdge. Because of incompatibility issues with prior versions of Tomcat, 11.6 (and earlier releases) of PAS for OpenEdge will not run on OpenEdge 11.7.

If you have pre-11.7 instances, you must do the following, after an OpenEdge 11.7 install, before you can run them:

1. Obtain a list of deployed instances:

```
proenv> pasman instances
```
The output will be a list of the alias names of registered instances and the full pathname of the instances' root directory. You will use this information in the following steps.

2. Stop all running instances. For example:

   proenv> pasman stop -I instance-name

3. Using the alias names, unregister each instance:

   proenv> pasman unregister alias-name

4. Using the alias name and the full pathname of the root directory, re-register each instance:

   proenv> pasman register alias-name pathname

   When you re-register the instance, configuration files are automatically updated to the Tomcat version that is compatible with 11.7.

Once the PAS for OpenEdge instance is updated and can run on 11.7, any deployed web applications must also be updated. Use Progress Developer Studio to perform the updates. See Progress Developer Studio for OpenEdge Migration Tasks on page 17 for more information.

**SSL Support Deprecated**

Due to the upgrade to Java 8 in OpenEdge 11.7, support for SSLv3 and other cipher suites (MD5 and RC5, for example) has been retired. Instead, support for TLS has been updated and replaces those retired technologies.

Nothing need be done to re-configure instances for TLS after completing an OpenEdge 11.7 and the instance migration described in Migrating existing PAS for OpenEdge instances on page 15. However, if you must revert to those retired technologies, you cannot use OpenEdge 11.7 or later releases.
Progress Developer Studio for OpenEdge
Migration Tasks

This section contains the steps to migrate the ABL Web App projects and Spring Security files to OpenEdge 11.7.

For details, see the following topics:

• Migrating an ABL Web App Project

Migrating an ABL Web App Project

If you have imported an ABL Web App project from OpenEdge 11.6 to the current release in the Progress Developer Studio for OpenEdge, you can also migrate the Spring Security files.

Note: You will not be allowed to export or deploy the 11.6 ABL Web App project until you migrate.

1. Right-click the ABL Web App project and select Progress OpenEdge > Migrate Spring Security Files. The Migrate Spring Security Files dialog box also opens.

Note: The dialog box also opens when you open an old workspace or import an old project to 11.7 workspace that contains ABL Web App projects.

2. Click OK.
Note: If you click **Cancel**, you must later follow the previous step to migrate.

The Spring Security files in the current **WEB-INF** folder in the project directory are moved to the newly created **backup** folder. The migrated Spring Security files are placed in the **WEB-INF** folder.
Migrating OpenEdge Business Process Management

This chapter describes the steps to migrate existing OpenEdge Business processes to OpenEdge 11.7. It contains sections that you may use depending on whether you are a user of Savvion 7.6/8.0 or Progress OpenEdge 11.6.x and earlier releases.

**Note:** If you are a Savvion 7.6/8.0 user, you must first follow the instructions in Savvion 7.6.X/8.0 to Progress OpenEdge 11.3.0 Migration Process on page 19 and migrate to Progress OpenEdge 11.3. You must then follow the instructions in the Migrating from OpenEdge 11.6.x and earlier releases on page 41 and migrate to Progress OpenEdge 11.7. If you are already a user of Progress OpenEdge 11.6.x and earlier releases, you can proceed to Migrating from OpenEdge 11.6.x and earlier releases on page 41.

For details, see the following topics:

- Savvion 7.6.X/8.0 to Progress OpenEdge 11.3.0 Migration Process
- Migrating from OpenEdge 11.6.x and earlier releases

**Savvion 7.6.X/8.0 to Progress OpenEdge 11.3.0 Migration Process**

This section contains all the information you require to migrate your Savvion 7.6.X/8.0 sources to Progress OpenEdge 11.3.0. By migrating to OpenEdge 11.3.0, you can make use of a new set of Progress Developer Studio for OpenEdge functionalities.
For more information about the new functionalities and product-release related information, see the OpenEdge guides and the release notes.

**Before you begin**

Do the following before beginning the Savvion 7.6.x/8.0 to OpenEdge 11.3.0 migration process:

1. Take a backup of the following:
   a) Progress Developer Studio for OpenEdge workspace
      Manually copy the BPM Studio workspace folder to any another location in your computer.
   b) OpenEdge Database
      Before you start the backup process, stop the AppServers/JBOSS Server and the database by using stopEjbServer.sh, stopPortalServer.sh, and stopsbmb.sh respectively.
      Your schema can become corrupt during the conversion or the migration process. If the conversion/migration fails, your database cannot be recovered. Therefore, you must take a backup of your database if you have to restore the database and begin the conversion/migration again.
      
      
      ```
      probkup Db-nameDevice-name
      ```

      *Db-name*

      Name of your database.

      *Device-name*

      Directory structure of the location where you want to back up the data.

      For example, if your *Db-name* is sbmdb and your *Device-name* is sbmdb.bkp file, then you must execute the following `probkup` command:

      ```
      probkup sbmdb sbmdb.bkp
      ```

      For more information on using the probkup utility, see *Progress OpenEdge Data Management: Database Administration*.

   2. Export the following:
      a) All the Savvion 7.6.x/8.0 projects and their artifacts.
      In Savvion BPM Studio, select **File > Export > General > Archive File** to export all your OpenEdge BPM projects as an archive file to your computer.
      As part of the migration, the archive file is imported to OpenEdge 11.3.0.

      b) Savvion 7.6.x/8.0 Business Objects
         - In Savvion BPM Studio, select **OpenEdge > Tools > Business Objects**. The **Business Object Manager/Channel Manager/Message Manager** dialog box appears.
         - Select the available Business Objects, and then click **Export**. The file system dialog box appears.
Specify the location in which you want to store your Business Objects, and then click OK.

Perform the above three steps to export Channels and Messages using the options **OpenEdge > Tools > Business Objects** and **OpenEdge > Tools > Channels** respectively.

3. The files that you store in your file system are imported to OpenEdge Developer Studio 11.3.0 as part of the migration. Ensure that BizStore and BizPulse process all the events; the values for last EventID in the BIZSTOREEVENTCOUNTER and BIZPULSEEVENTCOUNTER tables must match the maximum value of EventID in the BIZEVENT table.

4. When you install Target Progress Developer Studio for OpenEdge, specify the same administrator user name and password that you used in the Source BusinessManager (BM) installation. If you install the Target BusinessManager with a different administrator user name then, after the Target BM installation is complete, open the **User Configuration** tool and specify the Source BM administrator user and password for BusinessManager Admin Credentials.

5. For proper migration, both Source BM and Target BM must be configured with the same User Management Realm. If not, all the BusinessManager users will not be migrated. Your BusinessManager administrator must create the users.

6. If you are migrating from BusinessManager 7.6.3, run `setupsbm -c ProcessViews`. Ignore this step, if you performed it as a part of the BusinessManager 7.6.3 installation.

7. Back up the Source BM database.

8. Back up the file system of BusinessManager installation folder.

9. Install Target BM with a fresh database instance. Do not execute repository setup from the **First steps console** (or using the `setupsbm` command) with the Target BM installation. Ensure that no database objects (such as table and sequence) exist in this instance.

10. Perform an export dump of the Source BM database. Import it to the database instance that is configured with Target BM.

11. Ensure that there are no adapters in active state in the Source BM.

12. Ensure that all the JMS messages in BusinessManager specific Topics and Queues are processed in Source BM.

13. After importing the database dump of Source BM to the database instance of Target BM, drop the following application-server-specific tables into the migrated database manually.

   In the case of JBoss EAP 5.1, drop the following tables into the migrated database:

   ```
   JBM_COUNTER, JBM_DUAL, JBM_ID_CACHE, JBM_MSG, JBM_MSG_REF, JBM_POSTOFFICE, JBM_ROLE, JBM_TX, JBM_USER
   ```

**Points to remember**

- **OEBPM 11.3.0** does not support migration of live instances. Therefore, you must ensure that all active instances are completed and that the servers are stopped before migration.

- You cannot migrate a previous non-OpenEdge database to an OpenEdge database. You can only migrate a previous version of an OpenEdge database to an OpenEdge 11.3.0 database.

- You cannot migrate a previous non-JBOSS server to a JBOSS server. OpenEdge 11.3.0 only supports JBOSS server.

- OpenEdge 11.3.0 does not support BusinessManager clusters.
• Process Asset Manager (PAM) is deprecated in OpenEdge 11.3.0. If you have any process or artifacts in PAM, you must export them to the local file system using your existing BPM Studio before migrating to OpenEdge 11.3.0. Artifacts that are exported to the local file system can be re-imported to Progress Developer Studio for OpenEdge 11.3.0.

• BizStore data is not migrated.

• ABL Business Objects in Savvion 7.6.x are automatically migrated to the corresponding OpenEdge 11.3.0 Native datatypes (NDT). Savvion data types from Savvion 7.6.x/8.0 to OpenEdge 11.3.0 must be manually mapped as per data type mapping:

<table>
<thead>
<tr>
<th>Savvion datatype</th>
<th>ABL datatype</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>CHARACTER</td>
</tr>
<tr>
<td>Date</td>
<td>DATETIME-TZ</td>
</tr>
<tr>
<td>Boolean</td>
<td>LOGICAL</td>
</tr>
<tr>
<td>Number(Long)</td>
<td>INT64</td>
</tr>
<tr>
<td>Number</td>
<td>INTEGER</td>
</tr>
<tr>
<td>Number(Decimal)</td>
<td>DECIMAL</td>
</tr>
<tr>
<td>Number(Double)</td>
<td>Number(Double)</td>
</tr>
<tr>
<td>Map</td>
<td>MAP</td>
</tr>
<tr>
<td>URL</td>
<td>CHARACTER</td>
</tr>
<tr>
<td>List</td>
<td>LIST</td>
</tr>
<tr>
<td>Document</td>
<td>CHARACTER</td>
</tr>
<tr>
<td>XML</td>
<td>Not supported</td>
</tr>
</tbody>
</table>

• While migrating from Savvion 7.6/8.0 to Progress OpenEdge 11.3.0, the file names in the location `<Progress_Savvion>\SBM\conf DLC\oebpm\server\conf` change. Remember to update the file names wherever they are referred in your code. A few examples of modified file names are provided in the following table:

<table>
<thead>
<tr>
<th>Savvion datatype</th>
<th>ABL datatype</th>
</tr>
</thead>
<tbody>
<tr>
<td>blscheduler.properties</td>
<td>bpsscheduler.properties</td>
</tr>
<tr>
<td>bmaspect.properties</td>
<td>bpsaspect.properties</td>
</tr>
<tr>
<td>bmxsrconfig.xml</td>
<td>oebpsxrconfig.xml</td>
</tr>
<tr>
<td>bmxssconfig.xml</td>
<td>oebpssxssconfig.xml</td>
</tr>
<tr>
<td>sbmclient.conf</td>
<td>oebpmsclient.conf</td>
</tr>
</tbody>
</table>
Migrating to OpenEdge 11.3.0

Perform the following Savvion 7.6.x/8.0 to OpenEdge 11.3.0 migration steps:

1. Database migration on page 23
2. Application migration on page 29
3. Custom JSP forms migration on page 33

**Note:** You must install Progress Developer Studio for OpenEdge 11.3.0 in the same computer that has Savvion 7.6.x/8.0 installed in it.

Database migration

The database migration steps enable you to migrate your previous version OpenEdge database to OpenEdge 11.3.0 database.

Perform the following to migrate your database from Savvion 7.6.x/8.0 to OpenEdge 11.3.0:

1. Converting an OpenEdge release 10 database to Release 11 database
2. Configuring the OpenEdge 11.3.0 database
3. Setting OpenEdge Business Process Server environment variables
4. Migrating BusinessManager servers
5. Starting the server

Converting an OpenEdge Release 10 database to Release 11 database

The `PROUTIL CONV1011` utility converts the schema of an OpenEdge Release 10 database to the schema of an OpenEdge Release 11.

**Note:** Before you start the conversion process, ensure that your database is not in use.

To convert the schema of an OpenEdge Release 10 database to the schema of an OpenEdge Release 11 database:

<table>
<thead>
<tr>
<th>Savvion datatype</th>
<th>ABL datatype</th>
</tr>
</thead>
<tbody>
<tr>
<td>sbm.conf</td>
<td>oebps.conf</td>
</tr>
<tr>
<td>sbmdb.properties</td>
<td>oebpsdb.properties</td>
</tr>
<tr>
<td>sbmlog.conf</td>
<td>oebpslog.conf</td>
</tr>
<tr>
<td>bmwebsecurity.conf</td>
<td>oebpswebsecurity.conf</td>
</tr>
</tbody>
</table>
1. Truncate your before-image file with Release 10 PROUTIL TRUNCATE BI.

```
proutil Db-name -C truncate bi
```

*Db-name*

Name of your database.

For more information about using the `proutil` utility, see *Progress OpenEdge Data Management: Database Management*.

2. You can disable after-imaging, if it is enabled, with Release 10 RFUTIL AIMAGE END.

```
proutil Db-name -C aimage end
```

*Db-name*

Name of your database.

**Note:** If OpenEdge Replication is enabled, you must disable it before you can disable after-imaging. For information on Replication, see *Progress OpenEdge Replication: User Guide*.

3. Disable two-phase commit, if it is enabled, with Release 10 PROUTIL 2PHASE END.

```
proutil Db-name -C 2phase end
```

*Db-name*

Name of your database.

4. Copy the release 10 database to the OpenEdge Release 11 database folder.

If the two databases are on the same computer, you must execute the following command:

```
cp -R <source - Release 10 database location><destination - Release 11 database location>
```

The `cp` command copies the source directory to the destination directory.

5. Run the conversion utility to convert your database, as shown:

```
proutil Db-name -C conv1011
```
Db-name

Name of your database.

conv1011

Checks your database during pre-scan process. If the pre-scan encounters any problems that would prevent a successful conversion, the problems are logged and the convert utility exits without making any changes to the database. (Your database remains a valid OpenEdge Release 10 database.)

6. Optionally, perform an OpenEdge Release 11 database back up using the probkup utility.

```
probkup Db-name Device-name
```

Db-name

Name of your database.

Device-name

Location where you want to back up the data.

The above command performs a backup. For more information on using the probkup utility, see Progress OpenEdge Data Management: Database Administration.

You should back up your OpenEdge Release 11 database in case it becomes damaged.

**Configuring the OpenEdge 11.3.0 database**

After converting your OpenEdge Release 10 database to an OpenEdge 11 database, you must configure your OpenEdge 11 for the OpenEdge Developer Studio 11.3.0 installation.

To configure the OpenEdge 11 database:

1. Go to the database directory. Open a text file and insert the following content in it:

```
-db oebps
-s 8910
-n 150
-Mn 5
-Mi 1
-bibufs 25
-L 32000
-B 10000
-schlockwg
-SQLStmtCache 200
```

Name the text file oebps.pf and then save it.
2. Open a text file and insert the following content in it:

```sql
PROSQL_LOCKWAIT_TIMEOUT=302
sql_env
proserve -pf oebps.pf
```

Name the text file `db.cmd` (for Windows) or `db.sh` (for UNIX) and then save it.

3. Start the OpenEdge 11.3.0 database by executing the following command:

```
startdb.sh
```

**Note:** In UNIX, you must change the mode and then execute the `startdb.sh` command.

4. Configure the database using the **Database Configuration Tool** wizard:

- In Windows, select **Start > All Programs > Progress > OpenEdge 11.3 > BP Server > Tool > Database Configuration Tool.**
- In UNIX, execute the following script available at `$DLC/oebpm/server/bin`:

```
DBConfigTool.sh
```

5. Follow the **Database Configuration Tool** wizard to complete configuring the OpenEdge 11.3.0 database. Ensure that the Database name is the same as what you chose earlier, that is, oebps.

For more information about the options in the wizard, see the **Database Configuration Tool** section in *Progress OpenEdge Business Process Server: Server Administrator’s Guide.*

### Using the Data Migration tool

Perform the following to migrate your data from Savvion 7.6.x/8.0 to OpenEdge 11.3.0:

1. From `$DLC/oebpm/server/migration` (on UNIX) or `%DLC%\oebpm\server\migration` (on Windows), extract the OpenEdge BPM Data Migration Utility, `PROGRESS_OEBPM_DATA_MIGRATION_UTILITIES_ALL.zip` file, to a new folder on the computer on which OpenEdge 11.3.0 is installed. Then extract `PROGRESS_OEBPM_DATAMIGRATION_7.6.X_OR_8.X_TO_11.3.zip` in the same folder.

2. Open the `PROGRESS_OEBPM_DATAMIGRATION_7.6.X_OR_8.X_TO_11.3\bin` folder of the migration utility and set the environment variables in the `setenv.cmd/sh` file.

   Use the forward slash (`/`) for specifying folder paths.
### Property name | Description | Expected value
--- | --- | ---
JAVA_HOME | Specify the location of the JRE/JDK installation folder. | —
OEBPS_TARGET_HOME | Specify the Business Process Server (BP Server location) | $DLC\oebpm\server
OEBPS_SOURCE_VERSION | Specify the version of the source Business Manager (BM) | 7.6.x or 8.0
MIGRATION_HOME | Specify the location of the migration tool | $DLC\oebpm\server\migration

3. Open the PROGRESS OEBPM DATAMIGRATION 7.6.X OR 8.X TO 11.3\conf folder of the migration utility and set the environment variables in the migration.properties file.

| Property name | Description | Expected value
--- | --- | ---
oebps.target.home | Specify the location of the OpenEdge Business Process Server. | $DLC\oebpm\server
oebps.target.version | Specify the OpenEdge Business Process Server versions. | 11.3ALPHA
oebps.target.webapp.folder | Specify the location of webapp folder of your Business Process Server. | $DLC\oebpm\jboss\webapps
oebps.target.admin.userid | Specify the user name of the OpenEdge database administrator. | admin
oebps.source.version | Specify the version of the source business manager. | 7.6.x or 8.0
oebps.migration.home | Specify the location of the migration tool. | $DLC/oebpm/server/migration
oebps.migration.continueonerror | Retain false if you want migration to discontinue if an error occurs. The default value is false. | false or true
oebps.migration.verbose | Specify true if you want verbose logs during migration. | false or true
oebps.db.provider | Specify the database provider. | openedge
oebps.db.driver | Specify the name of the OpenEdge driver. | com.ddtek.jdbc.openedge.OpenEdgeDriver
oebps.db.multibyteschema | Specify true for multi-byte language support. | false or true
<table>
<thead>
<tr>
<th>Property name</th>
<th>Description</th>
<th>Expected value</th>
</tr>
</thead>
<tbody>
<tr>
<td>oebps.db.target.url</td>
<td>Specify the database URL.</td>
<td>jdbc:datadirect:openedge://localhost:&lt;port-number&gt;;DatabaseName=sbmdb</td>
</tr>
<tr>
<td>oebps.db.target.user</td>
<td>Specify the name of your database user, which you specified when installing and configuring OpenEdge Database during the OpenEdge 11.3 installation. By default, the name is dmadmin.</td>
<td>—</td>
</tr>
<tr>
<td>oebps.db.target.password</td>
<td>Specify the password of the database user, which you specified when installing and configuring the OpenEdge Database during the OpenEdge 11.3 installation. By default, the name is dbadmin.</td>
<td>—</td>
</tr>
</tbody>
</table>

4. If you have any deployed applications in Savvion 7.6.x/8.0, then you must copy them to the corresponding OpenEdge 11.3.0 installation. To copy your installation application to OpenEdge 11.3.0:

   a) Open the SBM_HOME\ebmsapps folder of your Savvion 7.6.x/8.0 installation. Copy all the application folders except Com, Common, OpenEdgeDDLOperationHandler and any other product shipped folders to the corresponding OpenEdge 11.3.0 installation, <drive>:\Progress\OpenEdge\oebpm\ebmsapps.

   Note: If you have defined your BPM application specific files and folders in the Common folder of your Savvion 7.6.x/8.0 installation, you must copy these artifacts to your OpenEdge 11.3.0 installation in <drive>:\Progress\OpenEdge\oebpm\Common.

   b) Open the SBM_HOME\webapps\deploy\sbm.war\ebmsapps folder of your Savvion 7.6.x/8.0 installation. Copy all the application folders except Com, Common, and OpenEdgeDDLOperationHandler to the corresponding OpenEdge 11.3.0 installation, <drive>:\Progress\OpenEdge\oebpm\jboss\webapps\deploy\sbm.war\ebmsapps.

   Note: Ensure that the Com, Common, and OpenEdgeDDLOperationHandler folders in the OpenEdge 11.3.0 installation are not replaced with the corresponding Savvion 7.6.x/8.0 folders.

5. If any application have rules, you must compile the application using the RuleCompiler utility, as below:

   a) At the command prompt, type cd OEBPS_TARGET_HOME\bin, and then press ENTER.

   b) At the <drive>:\OEBPS_TARGET_HOME\bin> prompt, type RuleCompiler -a AppName -blapi_lib -all, and then press ENTER.

6. Open the PROGRESS_OEBPM_DATAMIGRATION_7.6.X OR 8.X TO 11.3\bin folder of the migration utility and execute oebpsmigration.cmd/sh. If the migration is unsuccessful, view the error message and troubleshoot accordingly. Optionally, you can also view the success/errors/warnings in the migration.log file in the directory OEBPS_HOME\logs.
Starting the servers

To continue with application migration, start the application servers (EJB and Portal Server). In Windows, use the `Start > All Programs > Progress > OpenEdge 11.3 > BP Server > First Steps Console` to start and stop the servers.

Application migration

Use the Developer Studio to migrate the Source BPM Studio applications to the Target Developer studio applications.

The application migration is applicable to the following scenarios:

- From Savvion BPM Studio 7.6.x to Developer Studio 11.3.0. In this case, Savvion BPM Studio 7.6.x is the Source and Developer Studio 11.3.0 is the Target.
- From Savvion BPM Studio 8.0 to Developer Studio 11.3.0. In this case, Savvion BPM Studio 8.0 is the Source and Developer Studio 11.3.0 is the Target.

To migrate the Savvion BPM Studio applications to Developer Studio 11.3.0:

1. Open `Start > All Programs > Progress > OpenEdge 11.3 > Developer Studio - Clean`. Developer Studio appears.

2. To migrate workspace to OpenEdge 11.3.0:

   - To migrate your Savvion 7.6.x workspace to OpenEdge 11.3.0 workspace, do the following:
     1. Open `migrateWorkspace.bat` from `<DLC_HOME>/oebpm/studio/migration/76to113` (typically, the OpenEdge installation location, `DLC_HOME`, is `<drive>:\progress\Openedge`).
     2. Specify values for the following variables:

     | Variable                  | Description                                                                 |
     |---------------------------|------------------------------------------------------------------------------|
     | SAVVION_BPMSTUDIO_WORKSPACE | Specifies the Savvion BPM Studio workspace directory.                       |
     | OEBPM_WORKSPACE            | Specifies the new Developer Studio workspace directory that you use as the default workspace for OpenEdge 11.3.0. |
     | OE_DLC_HOEME               | Specifies the Progress Developer Studio for OpenEdge 11.3.0 installation directory. For example, C:\Progress\OpenEdge. |
     | PDS_HOME                   | Specifies the Eclipse installation directory. For example, C:\Progress\OpenEdge\oeide\eclipse. |

     3. Run the `migrateworkspace.bat` file.

   - To migrate your Savvion 8.0 workspace to OpenEdge 11.3.0 workspace, do the following:
1. Open `migrateWorkspace.bat` from `<DLC_HOME>/oebpm/studio/migration/80to113` (typically, the OpenEdge installation location, `DLC_HOME`, is `<drive>:\progress\Openedge\`).

2. Specify values for the following variables:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAVVION_BPMSTUDIO_WORKSPACE</td>
<td>Specifies the Savvion BPM Studio workspace directory.</td>
</tr>
<tr>
<td>OEBPM_WORKSPACE</td>
<td>Specifies the new Developer Studio workspace directory that you use as the default workspace for OpenEdge 11.3.0.</td>
</tr>
<tr>
<td>OE_DLC_HOME</td>
<td>Specifies the Progress Developer Studio for OpenEdge 11.3.0 installation directory. For example, C:\Progress\OpenEdge.</td>
</tr>
<tr>
<td>PDS_HOME</td>
<td>Specifies the Eclipse installation directory. For example, C:\Progress\OpenEdge\oeide\eclipse.</td>
</tr>
</tbody>
</table>

3. Run the `migrateworkspace.bat` file.

3. Use File > Import > Archive File to import the archive file that you exported from Savvion 7.6.x/8.0.

Importing a project into OpenEdge 11.3.0 automatically builds those projects. However, before you can publish the imported projects.

After you import your projects, for each project you must manually perform the following:

- Users who want to migrate their simulation projects from Savvion 7.6.0 to Progress OpenEdge 11.3.0 should first create a simulation project after installing OpenEdge 11.3.0 and copy the js folder from the Savvion directory `${WORKSPACE}\com.savvion.studio\simulation\reports` and paste it in the `{Simulation project}/reports` folder.

**Note:** This folder will be available in Progress OpenEdge 11.3.0 workspace only after a sample simulation project is created. For instructions on how to create a simulation project, refer to Creating a Simulation Project in *OpenEdge Getting Started: Developing BPM Applications with Developer Studio*.

- Import all the Business Objects, Channels, and Messages

  **Note:** If you have migrated your Savvion 7.6.x/8.0 workspace to OpenEdge 11.3.0 as per Step 2 on page 29, you do not need to manually import the Business Objects, Channels, and Messages. Proceed with Convert to Multiple Process project.

Use the Developer Studio menu bar options, **OpenEdge** > **Tools** > **Business Objects/Channels/Messages**, and import Business Objects to Developer Studio 11.3.0.

- Convert to Multiple Process project

Before publishing projects imported from BPM Studio 7.6.x to Developer Studio 11.3.0, you must manually convert each single-process project into a multiple-process project. To convert the projects, do the following:
Right-click the project, and select **Progress OpenEdge > Move and Convert to a Multiple Process BPM Project**. Follow the wizard to convert the project to a multiple-process BPM Project.

- **Migrate savvion datatypes to OpenEdge ABL datatypes**

Savvion datatypes from Savvion 7.6.x/8.0 are deprecated in OpenEdge 11.3.0. Therefore, when you open your Business processes (SPT) and custom adapters (Java/JavaScript) in OpenEdge 11.3.0, in the **Problems** view, you receive warnings about the usage of Savvion datatypes in the SPT files.

**Note:** You can access the **Problems** view using the menu option, **Window > Show views > Others... > Progress OpenEdge Editor > Problems**.

Although Savvion datatypes continue to work in OpenEdge 11.3.0, Progress Software recommends that you migrate Savvion datatypes to OpenEdge NDT or ABL datatypes to fix any warnings/errors that you might receive. Progress Software recommends that you migrate all the Savvion datatypes to OpenEdge NDT types as future product releases are likely to support only the OpenEdge NDT datatypes. Refer to the table in **Points to remember** on page 21 to understand the mapping between Savvion and ABL datatypes.

- **Update the Managed adapters in your business processes (SPT files)**

- **Migrate unsupported adapters to custom adapters:** Your projects cannot be published if they consist of unsupported adapters. Therefore, for every project imported to OpenEdge 11.3.0, you must replace any unsupported adapters in the project with the custom adapters.

- **In your business process (SPT files), all the unsupported adapters appear with a cross symbol.**

**Note:** If you attempt to publish a business process with any unsupported adapter, you will receive error messages regarding the usage of unsupported adapters in the Problems view, which must be fixed before publishing a process.

You can access the Problems view using the menu option **Window > Show views > Others... > Progress OpenEdge Editor > Problems**.

- **Update all Managed adapters**

  For all supported managed adapters in your business processes (SPT), you must delete the **maps** folder in the adapter. When you build your project, an updated and OpenEdge 11.3.0-compatible **maps** folder is created automatically.

- **Update OpenEdge Managed Adapter’s configuration**

  If your business processes (SPTs) have OpenEdge a managed adapter in them, perform the following:

  1. Select the managed adapter, and then select the **Properties > Configuration > Configure...** button.

     The **Configuration** dialog box appears.

  2. Re-select the procedure (.p) file for the **Input file** field. Edit the **Procedure Details** section, if required.

  3. Change the value for Application service from **sbmbroker1** to **bpsbroker1**.

  4. Click OK.

  5. Select the menu option **Project > Build All** to rebuild the project.

**Note:** After changing the managed adapter configuration or any configurations in the project, you must build the project to generate the new artifacts.
• Edit rules in Java/JavaScript code

All your Java/JavaScript code continues to work in OpenEdge BPM 11.3.0. However, you can only edit the files using the GEL Editor but you cannot use any drag-and-drop functionality from the pallets such as the Data Operations pallet and the Value Expression pallet.

**Note:** You access the GEL editor to edit Java/JavaScript code using **Properties > Advanced > Add/View script.**

• Edit Business Metric rules

All your applications containing business metric rule continues to work in OpenEdge 11.3.0. However, an application containing business metric rule on one of the activity configured with more than one condition fails to publish with compilation errors. This occurs due to an extra "(" generated between the conditions by the rule wizard. You must remove the extra "(" from between the conditions to make the application compile and publish in OpenEdge 11.3.0. Perform the following:

1. Open the business metric rule file from `<project_name>\<process_template_name>\rules\<business_metric_rule_file.bps>`.

2. Search for the additional "(" present between the conditions. The below sample illustrates an extra "(" generated between two conditions (marked in bold):

```java
rule MyBusinessMetricRule_MYRULE activated by EVT_1 of BizLogic::PI_COMPLETED {WORKSTEPNAME : "Activity 1", PROCESSTEMPLATENAME : "doub_str"}
{
    val piName = toString(EVT_1.context.PROCESSINSTANCENAME);
    if ( (doub_str::BM_MyBusinessMetricRule_Activity_b1_data::metric_ip[1][1].num1_Sum >= 0 )
        and ( (doub_str::BM_MyBusinessMetricRule_Activity_b1_data::metric_ip[1][1].num1_Maximum >= 100 ))
    )
}
```

3. Delete the extra "(" and save the rules file.

4. Re-deploy the application.

• Rebuild BPM Forms

All imported BPM forms must be re-built using the menu option **Project > Build All** before you re-install the business process in OpenEdge 11.3.0.

Your projects can then be published using OpenEdge 11.3.0.

4. Deploy Web, Common Resources, and BPM projects. Progress Software recommends that you deploy these projects with the **Reinstall** option.

5. To migrate applications published as Web services:

   a) Copy the `SBM_SOURCE_HOME\webapps\deploy\sbm\<ApplicationName>.jws` file to the `OEBPS_JBOSS\webapps\deploy\sbm` folder.

   b) Copy the `SBM_SOURCE_HOME\webapps\deploy\sbm\wsdl\<ApplicationName>.wsdl` file to the `OEBPS_JBOSS\webapps\deploy\sbm\wsdl` folder.

   c) Copy the `SBM_SOURCE_HOME\webapps\deploy\sbm\WEB-INF\jwsClasses` folder to the `OEBPS_JBOSS\webapps\deploy\sbm\WEB-INF` folder.
Custom JSP forms migration

This section explains how to migrate custom presentation forms created in earlier version of OpenEdge BPM studio to Progress Developer Studio for OpenEdge 11.3. When custom forms from older versions are migrated to Progress Developer Studio for OpenEdge 11.3.0, the form presentation is not changed or upgraded per Progress Developer Studio for OpenEdge 11.3.0.

You must perform a few manual steps to use the custom forms created in older version to be used in OpenEdge 11.3.0.

Note: The manual steps provided are only for guidance and may require more or fewer changes depending on the type of customization done.

Sample code is provided for the following scenarios:

- Migrating BPM Studio 8.0 to Progress Developer Studio for OpenEdge 11.3.0 on page 33
- Migrating BPM Studio 7.6.3 to Progress Developer Studio for OpenEdge 11.3.0 on page 34
- Migrating BPM Studio 7.6.1 to Progress Developer Studio for OpenEdge 11.3.0 on page 36

Migrating BPM Studio 8.0 to Progress Developer Studio for OpenEdge 11.3.0

To migrate custom JSPs to OpenEdge 11.3.0, you must edit the JSPs as instructed in the following tables:

Note: If you copy-paste the code from this document into your custom JSP, you must ensure that the code does not contain erroneous alignment and spacing, which can result in syntax errors or unsuccessful custom JSP migration.

<table>
<thead>
<tr>
<th>Find and Replace...</th>
<th>With...</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;bizsolo:link rel=&quot;stylesheet&quot;/&gt;</td>
<td>&lt;bizsolo:link rel=&quot;stylesheet&quot; extJsVer=&quot;3&quot;/&gt;</td>
</tr>
<tr>
<td>function openDocAttWin( slotName, sesID, ptname, piname, docurl,</td>
<td>function openDocAttWin( slotName, sesID, ptname, piname, docurl,</td>
</tr>
</tbody>
</table>
All the ABL dataslots in Savvion 7.6.x/8.0 were Business objects, but when they are migrated to OpenEdge 11.3.0, they are transformed into OpenEdge native ABL datatypes. Therefore, the custom JSP forms with ABL dataslots created in Savvion 7.6.x/8.0 require the following changes in OpenEdge 11.3.0:

* Previously the values were accessed as %DATASLOT_NAME%.value. Beginning with OpenEdge 11.3.0, the dataslots must be referenced as %DATASLOT_NAME%.

Migrating BPM Studio 7.6.3 to Progress Developer Studio for OpenEdge 11.3.0

To migrate custom JSPs to OpenEdge 11.3.0, you must edit the JSPs as instructed in the following tables:
### Find and Replace...

<table>
<thead>
<tr>
<th>Original Code</th>
<th>Revised Code</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>function openDocAttWin( slotName, sesID, ptname, piname, docurl, docServer, readonly, ismultiline, appendwith, isStart )</code></td>
<td><code>function openDocAttWin( slotName, sesID, ptname, piname, docurl, docServer, readonly, ismultiline, appendwith, isStart, fiid )</code></td>
</tr>
<tr>
<td><code>&lt;%@ taglib uri=&quot;http://java.sun.com/jstl/core&quot; prefix=&quot;c&quot; %&gt;</code></td>
<td><code>&lt;%@ taglib uri=&quot;http://java.sun.com/jsp/jstl/core&quot; prefix=&quot;c&quot; %&gt;</code></td>
</tr>
<tr>
<td><code>&lt;bizsolo:link rel=&quot;stylesheet&quot;&gt;&lt;/bizsolo:link&gt;</code></td>
<td><code>&lt;bizsolo:link rel=&quot;stylesheet&quot; extJsVer=&quot;3&quot;&gt;&lt;/bizsolo:link&gt;</code></td>
</tr>
<tr>
<td><code>formWidgetHandler = new FormWidgetHandler(allWidgets,</code></td>
<td><code>formWidgetHandler = new FormWidgetHandler(allWidgets,</code></td>
</tr>
<tr>
<td><code>{processName:'doctst5in763',adapletCache:</code></td>
<td><code>{processName:'doctst5in763',adapletCache:</code></td>
</tr>
</tbody>
</table>

**Note:** If you copy-paste the code from this document into your custom JSP, you must ensure that the code does not contain erroneous alignment and spacing, which can result in syntax errors or unsuccessful custom JSP migration.
Migrating BPM Studio 7.6.1 to Progress Developer Studio for OpenEdge 11.3.0

To migrate custom JSPs to OpenEdge 11.3.0, you must edit the JSPs, as instructed in the following tables:

**Note:** If you copy-paste the code from this document into your custom JSP, you must ensure that the code does not contain erroneous alignment and spacing, which can result in syntax errors or unsuccessful custom JSP migration.

---

### Find and Replace...

<table>
<thead>
<tr>
<th>Find...</th>
<th>With...</th>
</tr>
</thead>
<tbody>
<tr>
<td>`{processName:'doctst5in763',adapletCache: {'user':''}}};&lt;%request.getParameter(BizSoloRequest.BSS_FIID)%&gt;});</td>
<td><code>{user:''}, 'catch(e){return false;}</code></td>
</tr>
</tbody>
</table>

### And add the following after it...

<table>
<thead>
<tr>
<th>Find...</th>
<th>And add the following after it...</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>param += '&amp;isPICreation=' + isStart;</code></td>
<td><code>param += '&amp;fiid=' + fiid;</code></td>
</tr>
<tr>
<td><code>&lt;input name=&quot;_ProcessTemplateName&quot; type=&quot;hidden&quot; value='&lt;%=bean.getPropString(&quot;ptName&quot;)%&gt;'&gt;</code></td>
<td><code>&lt;input name=&quot;&lt;%=BizSoloRequest.BSS_FIID%&gt;&quot; type=&quot;hidden&quot; value='&lt;%=request.getParameter(BizSoloRequest.BSS_FIID)%&gt;'&gt;</code></td>
</tr>
<tr>
<td>Find and Replace...</td>
<td>With...</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------</td>
</tr>
<tr>
<td>beans.*,java.util.Vector,java.util.Locale&quot; %&gt;</td>
<td>sbm.util.DatabaseMapping,java.util.Vector,java.util.Locale&quot; %&gt;</td>
</tr>
<tr>
<td><code>&lt;%@ taglib uri=&quot;http://java.sun.com/jstl/core&quot; prefix=&quot;c&quot; %&gt;</code></td>
<td><code>&lt;%@ taglib uri=&quot;http://java.sun.com/jsp/jstl/core&quot; prefix=&quot;c&quot; %&gt;</code></td>
</tr>
<tr>
<td><code>&lt;script language=&quot;JavaScript&quot; src=&quot;&lt;c:out value='${contextPath}'/&gt;dwr/engine.js&quot;&gt;&lt;/script&gt;</code></td>
<td><code>&lt;script language=&quot;JavaScript&quot; src=&quot;&lt;c:out value='${contextPath}'/&gt;dwr/engine.js&quot;&gt;&lt;/script&gt;</code></td>
</tr>
<tr>
<td><code>&lt;link rel=&quot;stylesheet&quot; type=&quot;text/css&quot; href=&quot;&lt;c:out value='${contextPath}'/&gt;bpmportal/javascript/ext/resources/css/xtheme-default.css&quot;&gt;</code></td>
<td><code>&lt;link rel=&quot;stylesheet&quot; type=&quot;text/css&quot; href=&quot;&lt;c:out value='${contextPath}'/&gt;bpmportal/css/&lt;%= bizManage.getTheme() %&gt;/bm-all.css&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;link rel=&quot;stylesheet&quot; type=&quot;text/css&quot; href=&quot;&lt;c:out value='${contextPath}'/&gt;bpmportal/css/theme01/bm-all.css&quot;&gt;</code></td>
<td><code>&lt;link rel=&quot;stylesheet&quot; type=&quot;text/css&quot; href=&quot;&lt;c:out value='${contextPath}'/&gt;bpmportal/css/&lt;%= bizManage.getTheme() %&gt;/bm-xml.css&quot;&gt;</code></td>
</tr>
<tr>
<td>function openDocAttWin(slotName,sesID, ptname, piname, docurl, docServer, readonly, ismultiline, appendwith, isStart)</td>
<td>function openDocAttWin(slotName,sesID, ptname, piname, docurl, docServer, readonly, ismultiline, appendwith, isStart,fiid )</td>
</tr>
</tbody>
</table>
| `<td class="ApSegTitle" align="center">`<bizsolo:getDS>` | `<td class="ApSegTitle" align="center">`<bizsolo:choose>`<bizsolo:when>`
### Chapter 3: Migrating OpenEdge Business Process Management

<table>
<thead>
<tr>
<th>Find and Replace...</th>
<th>With...</th>
</tr>
</thead>
<tbody>
<tr>
<td>name=&quot;workitemName&quot;&gt; &lt;/bizsolo:getDS&gt;&lt;/td&gt;</td>
<td>test='$=bean.getPropString (&quot;workitemName&quot;) != null $'&gt;&lt;bizsolo:getDS name=&quot;workitemName&quot;&gt;$&lt;/bizsolo:getDS&gt; &lt;/bizsolo:when&gt;<a href="">bizsolo:otherwise</a> $=% _PageName%$&lt;/bizsolo:otherwise&gt; &lt;/bizsolo:choose&gt;&lt;/td&gt;</td>
</tr>
</tbody>
</table>
| <td width="15%
class="ApSegGenLabel"><bizsolo:getLabel type="RESOURCE" name="BIZSITE_INSTRUCTION_LABEL"> \$/</bizsolo:getLabel></td> | <td width="85%
class="ApSegGenLabel"><bizsolo:getLabel type="RESOURCE" name="BIZSITE_INSTRUCTION_LABEL"> \$/</bizsolo:getLabel></td> |
| <div style="display:none" id="textField1Error"><div><font color="red"><span id="textField1ErrorMsg"></span><a href="#"onclick="textField1ErrorMsgClose();onclick="textField1ErrorMsgClose();return false;"><img border="0" src="
value='${contextPath}'/bpmportal/css/apptheme01/images/close.gif"></a></font></div></div> | <div style="display:none" id="textField1Error"><div><font color="red"><span id="textField1ErrorMsg"></span><a href="#"onclick="textField1ErrorMsgClose();onclick="textField1ErrorMsgClose();return false;"><img border="0" src="
value='${contextPath}'/bpmportal/css/apptheme01/images/close.gif"></a></font></div></div> |
| <div style="display:none" id="textField2Error"><div><font color="red"><span id="textField2ErrorMsg"></span><a href="#"onclick="textField2ErrorMsgClose();onclick="textField2ErrorMsgClose();return false;"><img border="0" src="
value='${contextPath}'/bpmportal/css/apptheme01/images/close.gif"></a></font></div></div> | <div style="display:none" id="textField2Error"><div><font color="red"><span id="textField2ErrorMsg"></span><a href="#"onclick="textField2ErrorMsgClose();onclick="textField2ErrorMsgClose();return false;"><img border="0" src="
value='${contextPath}'/bpmportal/css/apptheme01/images/close.gif"></a></font></div></div> |

OpenEdge Getting Started: Migrating to OpenEdge 11.7
Find and Replace... | With...
---|---
```html
href="javascript://" onClick="setUserControl(document.form.bizsite_assigneeName);searchUser()"<img width="16" height="16" border="0" title="Search" src="<c:out src="value='${contextPath}'/>bpmportal/css/apptheme01/images/icon_edit_user_search_single.gif">
```
```html
href="javascript://" onClick="setUserControl(document.form.bizsite_assigneeName);searchUser()"<img width="16" height="16" border="0" title="Search" src="<c:out value='${contextPath}'/>bmpportal/css/app%3BbizManage.getTheme() %>
```
```html
<formWidgetHandler = new FormWidgetHandler(allWidgets, {processName:'testapp76',adapletCache:...})
```
```html
formWidgetHandler = new FormWidgetHandler(allWidgets, {processName:'testapp76',adapletCache:...})
```
### Chapter 3: Migrating OpenEdge Business Process Management

#### Find and Replace...

<table>
<thead>
<tr>
<th>Original</th>
<th>Replaced</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>{processName:'testapp76',adapletCache: {'user':''})</code>;</td>
<td><code>{processName:'testapp76',adapletCache: {'user':''})</code>;</td>
</tr>
<tr>
<td>exception.caught(e){}</td>
<td>exception.caught(e){return false;}</td>
</tr>
</tbody>
</table>

#### Find and Replace...

<table>
<thead>
<tr>
<th>Original</th>
<th>Replaced</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;!-- Javascript --&gt; &lt;script language=&quot;JavaScript&quot;&gt; var getLocalizedString = parent.getLocalizedString; &lt;/script&gt;</code></td>
<td><code>&lt;!-- Javascript --&gt; &lt;script language=&quot;JavaScript&quot;&gt; var getLocalizedString = parent.getLocalizedString; &lt;/script&gt;</code></td>
</tr>
<tr>
<td><code>&lt;script language=&quot;JavaScript&quot; src=&quot;&lt;c:out value='${contextPath}'/&gt;bpmportal/javascript/fvalidate/fValidate.validators.js&quot;&gt;&lt;/script&gt;</code></td>
<td><code>&lt;script language=&quot;JavaScript&quot; src=&quot;&lt;c:out value='${contextPath}'/&gt;bpmportal/javascript/fvalidate/fValidate.validators.js&quot;&gt;&lt;/script&gt;</code></td>
</tr>
<tr>
<td><code>&lt;script language=&quot;JavaScript&quot; src=&quot;&lt;c:out value='${contextPath}'/&gt;bpmportal/javascript/ux/fileuploadfield/</code></td>
<td><code>&lt;script language=&quot;JavaScript&quot; src=&quot;&lt;c:out value='${contextPath}'/&gt;bpmportal/javascript/ux/fileuploadfield/</code></td>
</tr>
<tr>
<td>Find and Replace...</td>
<td>With...</td>
</tr>
<tr>
<td>-----------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>javascript/...&gt;</td>
<td>FileUploadField.js&quot;&gt; &lt;/script&gt;</td>
</tr>
<tr>
<td>&lt;/script&gt;</td>
<td>&lt;script language=&quot;JavaScript&quot;</td>
</tr>
<tr>
<td></td>
<td>src=&quot;&lt;c:out value='${contextPath}'/&gt;</td>
</tr>
<tr>
<td></td>
<td>bpmportal/javascript/bm/common/bmfield.js&quot;</td>
</tr>
<tr>
<td></td>
<td>&lt;script language=&quot;JavaScript&quot;</td>
</tr>
<tr>
<td></td>
<td>src=&quot;&lt;c:out value='${contextPath}'/&gt;</td>
</tr>
<tr>
<td></td>
<td>bpmportal/javascript/fileupload.js&quot;</td>
</tr>
<tr>
<td></td>
<td>&lt;/script&gt;</td>
</tr>
<tr>
<td>param += '&amp;isPICreation=' + isStart;</td>
<td>param += '&amp;fiid=' + fiid;</td>
</tr>
<tr>
<td></td>
<td>&lt;input name=&quot;_ProcessTemplateName&quot; type=&quot;hidden&quot; value='\&lt;%=bean.getPropString(&quot;ptName&quot;) %&gt;'/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;input name=&quot;&lt;%=BizSoloRequest.BSS_FIID%&gt;&quot; type=&quot;hidden&quot; value='\&lt;%=request.getParameter(BizSoloRequest.BSS_FIID)%&gt;'/&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Find...</th>
<th>And add the following before it...</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;&lt;script language=&quot;JavaScript&quot;</td>
<td>&lt;script&gt;DWREngine = dwr.engine;</td>
</tr>
<tr>
<td>src=&quot;&lt;c:out value='${contextPath}'/&gt;/</td>
<td>DWRUtil = dwr.util;&lt;/script&gt;</td>
</tr>
<tr>
<td>bpmportal/javascript/utilities.js&quot;&lt;/script&gt;</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Remove the following lines of code if the exist...</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;bizsolo:link rel=&quot;stylesheet&quot;&gt;&lt;/bizsolo:link&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;bizsolo:if test='%=bean.getPropString(&quot;workitemName&quot;) !=&quot;null&quot;;'&gt;&lt;/bizsolo:if&gt;</td>
<td></td>
</tr>
</tbody>
</table>

**Migrating from OpenEdge 11.6.x and earlier releases**

This section describes the migration tasks you must perform to migrate your business processes from OpenEdge 11.6.x and earlier releases to OpenEdge 11.7.
Before you begin

Do the following before beginning the migration process:

Take a backup of the following:

a) Progress Developer Studio for OpenEdge workspace

   Manually copy the BPM Studio workspace folder to any another location in your computer.

b) OpenEdge Database

   Before you start the backup process, stop the AppServers/JBOSS Server and the database by using
   stopEjbServer.sh, stopEjbServer.cmd/sh, stopPortalServer.sh, and stopsbmdb.sh
   respectively.

   Your schema can become corrupt during the conversion or the migration process. If the
   conversion/migration fails, your database cannot be recovered. Therefore, you must take a backup of
   your database if you have to restore the database and begin the conversion/migration again.

   \[ probkup \ \text{Db-name} \ \text{Device-name} \]

   \text{Db-name}

   Name of your database.

   \text{Device-name}

   Directory structure of the location where you want to back up the data.

   For example, if your \text{Db-name} is sbmdb and your \text{Device-name} is sbmdb.bkp file, then you must execute
   the following probkup command:

   \[ probkup \ sbmdb \ sbmdb.bkp \]

   For more information on using the probkup utility, see Progress OpenEdge Data Management: Database
   Administration.

UNIX platform

On a UNIX platform, if the following files exist on your computer:

1. Delete \text{<DLC>/oebpm/jboss/server.ejbserver/deploy/messaging/hsqldb-persistence-service.xml}
2. Delete \text{<DLC>/oebpm/jboss/server.ejbserver/deploy/cluster/deploy-hasingleton-jboss-beans.xml}
3. Delete \text{<DLC>/oebpm/jboss/server/portalserver/deploy/cluster/deploy-hasingleton-jboss-beans.xml}
4. Delete <DLC>/oebpm/jboss/server/portalserver/deploy/jeb3-connectors-jboss-beans.xml

5. Provide execute permissions to the <DLC>/oebpm/jboss/bin location using the following command:
   
   chmod 775 <DLC>/oebpm/jboss/bin

---

**Configuring the OpenEdge database**

After migrating to OpenEdge latest release, you must configure its corresponding OpenEdge database for the OpenEdge Developer Studio installation.

To configure the OpenEdge database for the latest release:

1. Start the existing OpenEdge database by executing the following command:
   
   ```
   startdb.sh
   ```

   **Note:** In UNIX, you must change the mode and then execute the `startdb.sh` command.

2. Follow the **Database Configuration Tool** wizard to complete configuring the OpenEdge database.

   Ensure that the Database name is the same as what you chose earlier, that is, `oebps`.

   For more information about the options in the wizard, see the **Database Configuration Tool** section in *Progress OpenEdge Business Process Server: Server Administrator's Guide*.

---

**Using the Data Migration Utility**

Perform the following to migrate your data from OpenEdge 11.6.x and earlier releases to OpenEdge 11.7.

1. From `$DLC/oebpm/server/migration` (on UNIX) or `%DLC%\oebpm\server\migration` (on Windows), extract the OpenEdge BPM Data Migration Utility, `PROGRESS_OEBPM_DATA_MIGRATION.Utility_ALL.zip` file, to a new folder on the computer on which the OpenEdge latest release is installed.

2. Extract `PROGRESS_OEBPM_DATAMIGRATION_11.3.x_TO_11.7.zip` in the same folder.

3. Open the `PROGRESS_OEBPM_DATAMIGRATION_11.3.x_TO_11.7.zip\bin` folder of the migration utility and set the environment variables in the `setenv.cmd/sh` file.

   Use the forward slash (`/`) for specifying folder paths.

<table>
<thead>
<tr>
<th>Property name</th>
<th>Description</th>
<th>Expected value</th>
</tr>
</thead>
<tbody>
<tr>
<td>JAVA_HOME</td>
<td>Specify the location of the JRE/JDK installation folder.</td>
<td><code>$DLC/oebpm/jdk</code></td>
</tr>
<tr>
<td>OEBPS_TARGET_HOME</td>
<td>Specify the Business Process Server (BP Server location)</td>
<td><code>$DLC/oebpm\server</code></td>
</tr>
<tr>
<td>Property name</td>
<td>Description</td>
<td>Expected value</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--------------------------------------------------</td>
<td>--------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>OEBPS_SOURCE_VERSION</td>
<td>Specify the version of the source Business Manager (BM)</td>
<td>The previous OpenEdge release 11.x.x version from which you want to migrate your data. For example, 11.5.</td>
</tr>
<tr>
<td>MIGRATION_HOME</td>
<td>Specify the location of the migration tool</td>
<td>$DLC\oebpm\server\migration</td>
</tr>
</tbody>
</table>

4. Open the `PROGRESS_OEBPM_DATAMIGRATION_11.3.x_TO_11.7.zip\conf` folder of the migration utility and set the environment variables in the `migration.properties` file.

<table>
<thead>
<tr>
<th>Property name</th>
<th>Description</th>
<th>Expected value</th>
</tr>
</thead>
<tbody>
<tr>
<td>oebps.target.version</td>
<td>Specify the OpenEdge Business Process Server versions.</td>
<td>The latest OpenEdge release version</td>
</tr>
<tr>
<td>oebps.migration.continueonerror</td>
<td>Retain false if you want migration to discontinue if an error occurs. The default value is false.</td>
<td>false or true</td>
</tr>
<tr>
<td>oebps.migration.verbose</td>
<td>Specify true if you want verbose logs during migration.</td>
<td>false or true</td>
</tr>
</tbody>
</table>

5. If you have any deployed applications in previous OpenEdge 11.x.x installation, then you must copy them to the corresponding OpenEdge latest release installation. To copy your installation application to OpenEdge latest release:

   a) Open the `SBM_HOME\ebmsapps` folder of your previous OpenEdge 11.x.x installation. Copy all the application folders except `Com`, `Common`, `OpenEdgeDDLOperationHandler` and any other product shipped folders to the corresponding OpenEdge latest release installation, `<drive>:\Progress\OpenEdge\oebpm\ebmsapps`.

   Note: If you have defined your BPM application specific files and folders in the `Common` folder of your previous OpenEdge 11.x.x installation, you must copy these artifacts to your OpenEdge latest release installation in `<drive>:\Progress\OpenEdge\oebpm\Common`.

   b) Open the `SBM_HOME\webapps\deploy\sbm.war\ebmsapps` folder of your previous OpenEdge 11.x.x installation. Copy all the application folders except `Com`, `Common`, and `OpenEdgeDDLOperationHandler` to the corresponding OpenEdge latest release installation, `<drive>:\Progress\OpenEdge\oebpm\jboss\webapps\deploy\sbm.war\ebmsapps`.

   Note: Ensure that any of the Web application data is copied from `<drive>:\progress\OpenEdge\oebpm\BPMWorkFlow` folder to `<drive>:\progress\OpenEdge\oebpm\BPMWebFlow`.

6. If any application have rules, you must compile the application using the RuleCompiler utility, as below:
a) At the command prompt, type `cd OEBPS_TARGET_HOME\bin`, and then press ENTER.

b) At the `<drive>:\OEBPS_TARGET_HOME\bin>` prompt, type `RuleCompiler -a AppName -blapi_lib -all`, and then press ENTER.

7. Open the `PROGRESS_OEBPM_DATAMIGRATION_11.3.x_TO_11.7.zip\bin` folder of the migration utility and execute `oebpsmigration.cmd/sh`.

If the migration is unsuccessful, view the error message and troubleshoot accordingly. Optionally, you can also view the success/errors/warnings in the `migration.log` file in the directory `OEBPS_HOME\logs`.

**Decision step migration**

Condition expressions generated for a Decision step for OpenEdge data types in OpenEdge 11.6.x and earlier releases must be regenerated with new expression formats when migrated to OpenEdge 11.7. The regenerated expressions must be available in the Business Process Server by redeploying the updated processes with the Refresh option (right-click the processes in the Servers view, and then select Refresh on the context menu).

The condition expressions generated for OpenEdge datatypes that are defined with the Basic option in a Decision step are automatically generated with the new expression formats when the condition expressions migrated from OpenEdge 11.6.x and earlier releases to OpenEdge 11.7. However, the condition expressions generated with the Advanced option in a Decision step must be regenerated manually when the expressions migrated to OpenEdge latest release.

In OpenEdge latest release, the `OpenEdgeDataTypeComparisonUtil` class has in-built static methods that enable you to modify condition expressions of a Decision step defined with the Advanced option. The `OpenEdgeDataTypeComparisonUtil` class contains the following static methods:

- `public static boolean isShorterThan(char1, char2)` - Returns a TRUE value if the first operand object is shorter than the second operand object.
- `public static boolean isLongerThan(char1, char2)` - Returns a TRUE value if the first operand object is longer than the second operand object.
- `public static boolean contains(CHARACTER char1, CHARACTER char2)` - Returns a TRUE value if char2 is a substring of char1.
- `public static boolean startsWith(CHARACTER char1, CHARACTER char2)` - Returns a TRUE value if char2 is a prefix of char1.

For more information on the `OpenEdgeDataTypeComparisonUtil` class, see the Progress OpenEdge Business Process Management documentation and release notes.

To migrate condition expressions of a Decision step defined with the Advanced option in OpenEdge 11.6.x and earlier releases to OpenEdge 11.7, you must edit the condition expressions for the OpenEdge dataslots as instructed in the following tables.

**For the CHARACTER dataslot**

- For the CHARACTER dataslot
Table 1: Editing a condition expression for the Character dataslot

<table>
<thead>
<tr>
<th>Operation</th>
<th>In earlier releases</th>
<th>In 11.7</th>
</tr>
</thead>
<tbody>
<tr>
<td>equals ^1</td>
<td>charDs1.equals(charDs2)</td>
<td>charDs1.equals(charDs2)</td>
</tr>
<tr>
<td>does not equal ^1</td>
<td>!charDs1.equals(charDs2)</td>
<td>!charDs1.equals(charDs2)</td>
</tr>
<tr>
<td>is shorter than ^1</td>
<td>CharDs1.length()&lt;CharDs2</td>
<td>com.progress.util.OpenEdgedataTypeComparisonUtil.isShorterThan(charDs1, charDs2)</td>
</tr>
<tr>
<td></td>
<td>(or) CharDs1.length()&lt;CharDs2.length()</td>
<td></td>
</tr>
<tr>
<td>is longer than ^1</td>
<td>CharDs1.length()&gt;CharDs2</td>
<td>com.progress.util.OpenEdgedataTypeComparisonUtil.isLongerThan(charDs1, charDs2)</td>
</tr>
<tr>
<td></td>
<td>(or) CharDs1.length()&gt;CharDs2.length()</td>
<td></td>
</tr>
<tr>
<td>is null ^1</td>
<td>charDs1==null</td>
<td>charDs1.isNull( )</td>
</tr>
<tr>
<td>is not null ^1</td>
<td>charDs1!=null</td>
<td>!charDs1.isNull( )</td>
</tr>
<tr>
<td>contains ^1</td>
<td>charDs1.indexOf(&quot;string&quot;)&gt;=0</td>
<td>com.progress.util.OpenEdgedataTypeComparisonUtil.contains(charDs1, CharDs2)</td>
</tr>
<tr>
<td>does not contain ^1</td>
<td>charDs1.indexOf(&quot;string&quot;)&lt;0</td>
<td>!com.progress.util.OpenEdgedataTypeComparisonUtil.contains(charDs1, CharDs2)</td>
</tr>
<tr>
<td>starts with ^1</td>
<td>charDs2.startsWith(&quot;string&quot;)</td>
<td>com.progress.util.OpenEdgedataTypeComparisonUtil.startsWith(charDs1, CharDs2)</td>
</tr>
<tr>
<td>does not start with ^1</td>
<td>!charDs2.startsWith(&quot;string&quot;)</td>
<td>!com.progress.util.OpenEdgedataTypeComparisonUtil.startsWith(charDs1, CharDs2)</td>
</tr>
<tr>
<td>is unknown ^1</td>
<td>--</td>
<td>charDs1.isNull( )</td>
</tr>
<tr>
<td>is not unknown ^1</td>
<td>--</td>
<td>!charDs1.isNull( )</td>
</tr>
</tbody>
</table>
For the INTEGER dataslot

- For the INTEGER dataslot:

<table>
<thead>
<tr>
<th>Operation</th>
<th>In earlier releases</th>
<th>In 11.7</th>
</tr>
</thead>
<tbody>
<tr>
<td>equals</td>
<td>myInt==urInt</td>
<td>integer1.equals(integer2)</td>
</tr>
<tr>
<td>does not equal</td>
<td>myInt!=urInt</td>
<td>!integer1.equals(integer2)</td>
</tr>
<tr>
<td>less than</td>
<td>myInt&lt;urInt</td>
<td>com.progress.util.OpenEdge</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DataTypeComparisonUtil</td>
</tr>
<tr>
<td></td>
<td></td>
<td>.isLesserThan(integer1, integer2)</td>
</tr>
<tr>
<td>less than or equal to</td>
<td>myInt&lt;=urInt</td>
<td>com.progress.util.OpenEdge</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DataTypeComparisonUtil</td>
</tr>
<tr>
<td></td>
<td></td>
<td>.isLessOrEq(integer1, integer2)</td>
</tr>
<tr>
<td>greater than</td>
<td>myInt&gt;urInt</td>
<td>com.progress.util.OpenEdge</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DataTypeComparisonUtil</td>
</tr>
<tr>
<td></td>
<td></td>
<td>.isGreaterThan(integer1, integer2)</td>
</tr>
<tr>
<td>greater than or equal to</td>
<td>myInt&gt;=urInt</td>
<td>com.progress.util.OpenEdge</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DataTypeComparisonUtil</td>
</tr>
<tr>
<td></td>
<td></td>
<td>.isGtOrEq(integer1, integer2)</td>
</tr>
<tr>
<td>is unknown</td>
<td>–</td>
<td>integer1.isNull( )</td>
</tr>
<tr>
<td>is not unknown</td>
<td>–</td>
<td>!integer1.isNull( )</td>
</tr>
</tbody>
</table>

For the INT64 dataslot

- For the INT64 dataslot

---

1. The String literal must be wrapped in the CHARACTER object. For example, `new com.progress.lang.Character("stringLiteral")`.

2. The Integer literal must be wrapped in the INTEGER object. For example, `com.progress.lang.Integer(integerLiteral)`. 
Table 3: Editing a condition expression for the INT64 datalot

<table>
<thead>
<tr>
<th>Operation</th>
<th>In earlier releases</th>
<th>In 11.7</th>
</tr>
</thead>
<tbody>
<tr>
<td>equals(^3)</td>
<td>int64_1==int64_2</td>
<td>int64_1.equals(int64_2)</td>
</tr>
<tr>
<td>does not equal(^3)</td>
<td>int64_1!=int64_2</td>
<td>!int64_1.equals(int64_2)</td>
</tr>
<tr>
<td>less than(^3)</td>
<td>int64_1&lt;int64_2</td>
<td>com.progress.util.OpenEdge</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DataTypeComparisonUtil.isLesserThan(int64_1, int64_2)</td>
</tr>
<tr>
<td>less than or equal to(^3)</td>
<td>int64_1&lt;=int64_2</td>
<td>com.progress.util.OpenEdge</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DataTypeComparisonUtil.isLessOrEq(int64_1, int64_2)</td>
</tr>
<tr>
<td>greater than(^3)</td>
<td>int64_1&gt;int64_2</td>
<td>com.progress.util.OpenEdge</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DataTypeComparisonUtil.isGreaterThan(int64_1, int64_2)</td>
</tr>
<tr>
<td>greater than or equal to(^3)</td>
<td>int64_1&gt;=int64_2</td>
<td>com.progress.util.OpenEdge</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DataTypeComparisonUtil.isGtOrEq(int64_1, int64_2)</td>
</tr>
<tr>
<td>is unknown(^3)</td>
<td>–</td>
<td>myInt64.isNull()</td>
</tr>
<tr>
<td>is not unknown(^3)</td>
<td>–</td>
<td>!myInt64.isNull()</td>
</tr>
</tbody>
</table>

For the DECIMAL dataset

- For the DECIMAL datalot

\(^3\) The integer literal must be wrapped in the INT64 object. For example, `com.progress.lang.Int64(longLiteral)`
Table 4: Editing a condition expression for the DECIMAL dataslot

<table>
<thead>
<tr>
<th>Operation</th>
<th>In earlier releases</th>
<th>In 11.7</th>
</tr>
</thead>
<tbody>
<tr>
<td>equals^4</td>
<td>jst.getDataSlotValue(&quot;myDec1&quot;) .getValue().equals(new java.math.BigDecimal(jst.getDataSlotValue(&quot;myDec2&quot;) .getValue().toPlainString()) .setScale(jst.getDataSlotValue(&quot;myDec1&quot;) .getValue().scale(), java.math.BigDecimal.ROUND_DOWN))</td>
<td>mydec1.equals(mydec2)</td>
</tr>
<tr>
<td>does not equal^4</td>
<td>!myDec1.equals(new java.math.BigDecimal(myDec2).setScale(myDec1.scale(), java.math.BigDecimal.ROUND_DOWN))</td>
<td>!mydec1.equals(mydec2)</td>
</tr>
<tr>
<td>less than^4</td>
<td>myDec1.compareTo(new java.math.BigDecimal(myDec2).setScale(myDec1.scale(), java.math.BigDecimal.ROUND_DOWN)) &lt; 0</td>
<td>com.progress.util.OpenEdgeData TypeComparisonUtil.isLesserThan(myDec1,myDec2)</td>
</tr>
<tr>
<td>less than or equal to^4</td>
<td>myDec1.compareTo(new java.math.BigDecimal(myDec2).setScale(myDec1.scale(), java.math.BigDecimal.ROUND_DOWN)) &lt;= 0</td>
<td>com.progress.util.OpenEdgeData TypeComparisonUtil.isLessOrEq(myDec1,myDec2)</td>
</tr>
<tr>
<td>greater than^4</td>
<td>myDec1.compareTo(new java.math.BigDecimal(myDec2).setScale(myDec1.scale(), java.math.BigDecimal.ROUND_DOWN)) &gt; 0</td>
<td>com.progress.util.OpenEdgeDataType ComparisonUtil.isGreaterThan(myDec1,myDec2)</td>
</tr>
<tr>
<td>greater than or equal to^4</td>
<td>myDec1.compareTo(new java.math.BigDecimal(myDec2).setScale(myDec1.scale(), java.math.BigDecimal.ROUND_DOWN)) &gt;= 0</td>
<td>com.progress.util.OpenEdgeDataType ComparisonUtil.isGtOrEq(MyDec1,MyDec2)</td>
</tr>
<tr>
<td>is unknown^4</td>
<td>myDec1==null</td>
<td>mydec1.isNull()</td>
</tr>
<tr>
<td>is not unknown^4</td>
<td>mydec1!=null</td>
<td>!mydec1.isNull()</td>
</tr>
</tbody>
</table>

For the LOGICAL dataslot

- For the LOGICAL dataslot
Table 5: Editing a condition expression for the LOGICAL dataslot

<table>
<thead>
<tr>
<th>Operation</th>
<th>In earlier releases</th>
<th>In 11.7</th>
</tr>
</thead>
<tbody>
<tr>
<td>equals&lt;sup&gt;5&lt;/sup&gt;</td>
<td>myLogic1==myLogic2</td>
<td>myLogic1.equals(MyLogic2)</td>
</tr>
<tr>
<td>does not equal&lt;sup&gt;5&lt;/sup&gt;</td>
<td>myLogic1!=myLogic2</td>
<td>!myLogic1.equals(MyLogic2)</td>
</tr>
<tr>
<td>is true&lt;sup&gt;5&lt;/sup&gt;</td>
<td>myLogic</td>
<td>myLogic.equals(new com. progress.lang.Logical(true))</td>
</tr>
<tr>
<td>is false&lt;sup&gt;5&lt;/sup&gt;</td>
<td>!myLogic</td>
<td>myLogic.equals(new com. progress.lang.Logical(false))</td>
</tr>
<tr>
<td>is unknown&lt;sup&gt;6&lt;/sup&gt;</td>
<td>myLogic==null</td>
<td>myLogic.isNull( )</td>
</tr>
<tr>
<td>is not unknown&lt;sup&gt;6&lt;/sup&gt;</td>
<td>myLogic!=null</td>
<td>!myLogic.isNull( )</td>
</tr>
</tbody>
</table>

For the DATETIME-TZ dataslot

- For the DATETIME-TZ dataslot

---

<sup>4</sup> The Decimal literal must be wrapped in the DECIMAL object. For example, `com.progress.lang.Decimal(new java.math.BigDecimal("1234.34"))`.

<sup>5</sup> The Boolean literal must be wrapped in the LOGIC object. For example, `new com.progress.lang.Logical(true/false)`.

<sup>Note:</sup> You must use the constructor of the BigDecimal method, which takes a string of the numeric decimal as an argument, and/or initialize the object with a correct scale.
### Table 6: Editing a condition expression for the DATETIME-TZ dataslot

<table>
<thead>
<tr>
<th>Operation</th>
<th>In earlier releases</th>
<th>In 11.7</th>
</tr>
</thead>
<tbody>
<tr>
<td>equals(^6)</td>
<td><code>Mydtz1.compareTo(Mydtz2)==0</code></td>
<td><code>Mydtz1.equals(Mydtz2)</code></td>
</tr>
<tr>
<td>does not equal(^6)</td>
<td><code>Mydtz1.compareTo(Mydtz2)!=0</code></td>
<td><code>!Mydtz1.equals(Mydtz2)</code></td>
</tr>
<tr>
<td>is before(^6)</td>
<td><code>Mydtz1.compareTo(Mydtz2)&lt;0</code></td>
<td><code>com.progress.OpenEdgeDataType</code> &lt;br&gt;<code>ComparisonUtil.isBefore</code> &lt;br&gt;(<code>Mydtz1</code>, <code>Mydtz2</code>)</td>
</tr>
<tr>
<td>is before or equal to(^6)</td>
<td><code>Mydtz1.compareTo(Mydtz2)&lt;=0</code></td>
<td><code>com.progress.OpenEdgeDataType</code> &lt;br&gt;<code>ComparisonUtil.isBeforeOrEq</code> &lt;br&gt;(<code>Mydtz1</code>, <code>Mydtz2</code>)</td>
</tr>
<tr>
<td>is after(^6)</td>
<td><code>Mydtz1.compareTo(Mydtz2)&gt;0</code></td>
<td><code>com.progress.OpenEdgeDataType</code> &lt;br&gt;<code>ComparisonUtil.isAfter</code> &lt;br&gt;(<code>Mydtz1</code>, <code>Mydtz2</code>)</td>
</tr>
<tr>
<td>is after or equal to(^6)</td>
<td><code>Mydtz1.compareTo(Mydtz2)&gt;=0</code></td>
<td><code>com.progress.OpenEdgeDataType</code> &lt;br&gt;<code>ComparisonUtil.isAfterOrEq</code> &lt;br&gt;(<code>Mydtz1</code>, <code>Mydtz2</code>)</td>
</tr>
<tr>
<td>is unknown(^6)</td>
<td><code>–</code></td>
<td><code>Mydtz.isNull()</code></td>
</tr>
<tr>
<td>is not unknown(^6)</td>
<td><code>–</code></td>
<td><code>!Mydtz.isNull()</code></td>
</tr>
</tbody>
</table>

\(^6\) The DateTime literal must be wrapped in the DATETIME-TZ object. For example, new `com.progress.lang.DateTimeTz("2014-01-01T16:22:54.000+05:30")`.  

**Note:** The DateTime literal must be in OpenEdge format: `yyyyMMdd'-'HH:mm:ss.SSS('+' |'-')zz:zz`.  

---