TIBCO Business Studio™ - BPM Edition

Migration Guide

Version 5.5.0 | March 2024

Document Updated: July 2024

Contents

What is Changed?

The following functionalities have been changed in this release of TIBCO Business Studio - BPM Edition 5.5.0:

Feature	Description
BPM Projects	There are some changes to the BPM projects. • Previously, you could create projects for destinations other than ActiveMatrix BPM, as well as for other add-ons like simulation. This is no longer the case. There are no other destination environments. Therefore, configuration for destination environments is no longer shown. • The existing BPM Developer Project now changes to BPM Process Project. • You must have separate projects for business object models and organization models. **Note:** TIBCO Business Studio - BPM Edition provides a quick fix to move business object models or organization models into dedicated projects.
Web Services	ActiveMatrix BPM version 5.x onwards does not support: • Using web services with WSDL documents. You must convert web-service invocation tasks to REST service tasks that invoke equivalent REST services. The web-service invocation tasks to REST conversion is done using TIBCO ActiveMatrix BusinessWorks. Here are the Github samples. • Exposing service operations that external clients or other processes can invoke using WSDL documents. Any of your processes that were previously exposed as web services can now be called using the run-time public REST API. Content and configuration related to WSDL documents are removed on

Feature	Description
	importing pre-version 5.x ActiveMatrix BPM projects to version 5.x onwards. See WSDL Activities for more information.
	When migrating from 4.x projects to the TIBCO Business Studio - BPM Edition 5.5.0 version, converts all the SOAP Service Consumer Participants to REST Service Consumer Participants.
Invoke Business Process Send Tasks	In BPM 4.x, business service processes could invoke a business process by invoking the web-service exposed for the message start event in the business process. This was done using a send-task with a service type of **Invoke Business Process**.
	Web-services and message start events are no longer supported by BPM 5.x and therefore these send-task invocations are automatically migrated to equivalent asynchronous sub-process call tasks. You must change the business process message start event type to **Start Request Event**, so that it can be invoked as a sub-process. Data mappings are preserved. However, if the send-task formerly used JavaScript data mappings then these are converted to Data Mapper grammar mappings. In this case, any user-defined script mappings are preserved but must be modified to explicitly return a value from the script. Invocation of business processes directly via the web-service that is generated for message start events cannot be automatically migrated and you must convert these to asynchronous sub-process call tasks manually.
Data	Run-time business internal representation of data has changed from XML to JSON. That means XML is not supported as a mechanism for creating business object models.
	Process data fields are now wrapped in a `data` object. On import migration of 4.x to 5.x projects, process field references are automatically fixed wherever possible.
	Note: The XML Serialization methods in TIBCO Business Studio - BPM Edition 4.x are replaced by `bpm.scriptUtils.stringify` and `parseData` methods. For more information, see the bpm.script.Util section in the *Application Designer Guide.*

Feature	Description
	For more information, see Scripting.
Data mapping	As internal representation of data is changing from XML to JSON, data-mappings are being standardized for the data mapper grammar. All JavaScript grammar mappings for task types still supported in 5.x are automatically converted to data mapper grammar. **Note:** You can use pre-version 5.x TIBCO Business Studio - BPM Edition side by side with version 5.x TIBCO Business Studio - BPM Edition. This means you can look at your data mappings in an original project to help you remap your data in a new project. For more information, see Data Mapping .
Forms	Some API methods are different between pre-version 5.x and version 5.x. When importing pre-version 5.x projects to TIBCO Business Studio - BPM Edition, the API methods are converted. For more information, see the *TIBCO Business Studio™ - BPM Edition Forms User Guide*.
Scripts	As data internal representation is changing from XML to JSON, there are changes to the JavaScript capabilities. For example, multiple instances of data are now represented as arrays rather than specialized list objects. Many JavaScript capabilities have also been standardized between process scripts and form scripts. Factory methods have also changed. The ScriptUtil allows the conversion between the BOM Objects and the JSON strings. The DateTimeUtil allows you to convert DateTime to DateTimeTZ for both string and date. See Scripting.
Business Object Models	The following are changes to Business Object Models: • Existing Business Object Models that were generated from WSDL (previously in the **Generated Business Objects** folder) are automatically moved to a **Business Objects** folder. This is because

Feature	Description
	they might still be used in processes, even if they are not used to invoke WSDL services.
	• A new concept of a terminal state is introduced. All case classes must have a case state attribute. This attribute must nominate the case state enumerations that indicate when a case is complete. You must set a case to one of its terminal states when there is no more work to be done on it.
	• The Case Identifier attribute has changed. You can choose to generate case identifiers automatically, or enter them manually. If you choose **Auto**, the Case Identifier can include a prefix or a suffix. For example, UK-1234-A. See Case Identifiers.
	• Some primitive BOM types are no longer supported: Attachment, DateTime, Duration, ID, Integer, Object.
	• Class generalization is no longer supported. On import migration of 4.x to 5.x projects, attributes in a class that are inherited by a class generalization are pulled up into the sub-class.
Deployment	You no longer export your project as a Distributed Application Archive (DAA). Instead, you generate deployment artifacts Runtime Application Shared Concept (RASC) files. These files are deployed using the ActiveMatrix BPM Administrator. Generating deployment artifacts is now done using the **Deployment** menu when right-clicking a project in TIBCO Business Studio - BPM Edition. The project lifecycle of development, acceptance test, and deployment to production through to creation of the next draft version is managed through the options on this new menu. For more information, see the Project Lifecycle section of the *TIBCO Business Studio™ - BPM Edition, Application Designer's Guide.*

Importing Pre-Version 5.x Projects into TIBCO Business Studio - BPM Edition Version 5.x

Projects that were created in pre-version 5.x TIBCO Business Studio - BPM Edition can be imported into TIBCO Business Studio - BPM Edition version 5.x and newer.

Some features that were available in pre-version 5.x projects are no longer available in version 5.x (and newer). See What's Changed? Any features contained in your pre-version 5.x projects that are no longer supported can cause problem markers to appear in TIBCO Business Studio - BPM Edition. After you have imported your pre-version 5.x projects, you must perform some post migration tasks to fix these problem markers. See Post Migration Tasks.

When you are importing pre-version 5.x projects, the Import wizard determines which projects are dependencies of other projects, and assists you with importing those projects first. After importing the lower-level projects, any problem markers should be resolved before continuing to import the higher-level projects, as the higher-level projects depend on the lower-level projects operating properly.

Importing Pre-Version 5.x Projects

TIBCO Business Studio - BPM Edition contains an Import wizard that you can use to import pre-version 5.x projects into TIBCO Business Studio - BPM Edition version 5.x and newer.

When projects are imported into TIBCO Business Studio - BPM Edition, it is important that lower-level projects - which other projects are dependent on - are imported first. For instance, if you have a process project that is dependent on an organization model project, the organization model project must be imported first, then the process project can be imported.

The Import wizard in TIBCO Business Studio - BPM Edition assists in importing projects with dependencies by indicating that there are projects that require migration whose dependencies have not already been imported, and allowing you to filter these out of the selection. After the lower-level projects are imported, you use the Import wizard again to

import the projects whose dependent projects have already been imported. This process is then repeated for each tier of projects until all have been imported.

Procedure

1. Start TIBCO Business Studio - BPM Edition with a blank workspace.

2. Select **File > Import > Existing Studio Projects into Workspace**.

3. On the Import Projects dialog, select either **Select root directory** or **Select archive file**. Then browse to and select the project folder or archive file for the pre-version 5.x project.

   If the project folder or archive contains multiple projects with dependencies, a message is displayed stating that it cannot import a project unless its dependencies are imported first. For example:
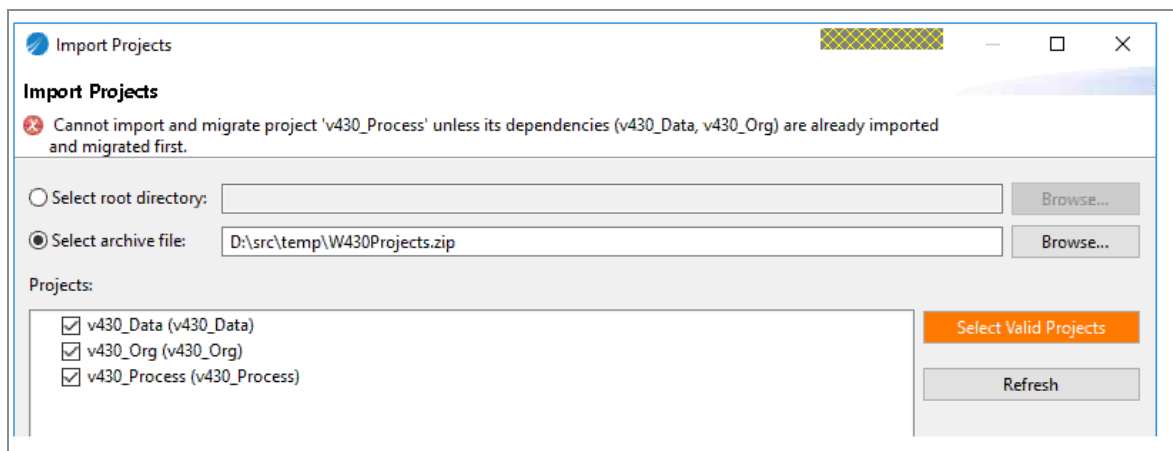
4. Click **Select Valid Projects**.

   The Import wizard deselects the projects that have dependencies.

5. Click **Finish**.

   The Import wizard imports the projects that were selected on the Import Projects dialog.

6. Resolve any issues that are flagged by problem markers.

   This can make the task of fixing problems more manageable by avoiding problems raised for higher-level projects that are actually caused by problems in lower-level projects. See Post Migration Tasks.

7. Repeat steps 2 - 6 to import all the projects in the archive.

Post migration tasks

After you have imported your pre-version 5.x TIBCO BPM Enterprise projects, any features contained in your projects that are no longer supported might have problem markers.

You must perform some post migration tasks to fix these problem markers.

Business Object Model

Pre-version 5.x ActiveMatrix BPM uses JavaScript representation for business object model data types.

Data Types

You can use the problem markers to help you fix any issues with your data type fields.

- Some primitive BOM types are no longer supported: Attachment, DateTime, Duration, ID, Integer, Object. Any references to these types cause problem markers.

- For simple type fields, the generic Object type is no longer supported for attributes. In JSON, use a text attribute and place your escaped JSON in the text attribute instead. In other words, this replaces XSD:any.

- Integers are now shown as a fixed point number with zero decimal places. Any integer types are automatically converted on import.

- There are two types of number:

    ○ Number.

    ○ Fixed point number.

    Both number types are implemented as JavaScript numbers with extension functions to help with various interactions. Fixed point numbers allow you to specify a fixed number of decimal places and the values assigned to these are rounded appropriately.

- Sub-packages are no longer supported. If your business object model has sub-packages, you must move all your classes up to the top-level package.

- For other restrictions, use the problem markers to help you reach a solution.

WSDL Activities

ActiveMatrix BPM version 5.x onwards does not invoke or expose service operations that external clients or other processes can invoke using WSDL documents. Content and configuration relating to WSDL documents are removed on importing pre-version 5.x ActiveMatrix BPM projects to version 5.x TIBCO Business Studio.

- TIBCO Business Studio version 5.x onwards processes do not support activities with a service type of **Web Service.**

    - To invoke business processes from business services or pageflow processes, replace the web service tasks with a call sub-process activity and configure the sub-process activity to be invoked asynchronously.

    - To invoke external clients, you must use REST services. To do this, on the **General** tab in the **Properties View** for your activity, set **Service Type** for your activity to **REST Service**.

- Incoming message activities (in other words, all those activities involved in exposing processes as web services) are no longer supported. Instead, you can use new incoming request activities (formerly receive task and intermediate untriggered or no-type events) in your process and trigger these using the generic runtime API services.

- Incoming request-with-reply patterns are not currently supported. As an alternative you might consider using an asynchronous send-two-ways pattern. The request is made from an external system via an Incoming request event (using the generic runtime API services), the 'reply' is implemented as a REST service send task that invokes a service in the external system to send the required reply data.

- Correlation is now supported using either process-instance-id (the same as pre-version 5.x) or business data. This means that you can now correlate certain incoming request event or task triggered by the Process REST API to an individual process instance by matching the correlation-data values in the input payload of the Process API with the correlation-data values in the interface properties associated with the incoming request event or task.
  See Controlling Flow from an External Application for more details.

Other Service Tasks

Some service types are no longer available for activities with a type of **Service Task**.

Service Type	Description
Java	You must use REST services. To do this, on the **General** tab in the **Properties View** for your activity, set **Service Type** for your activity to **REST Service**.
Document operation	You can control documents using the ActiveMatrix BPM Client or the generic runtime API.
Decision	You can use JavaScript, or in complex scenarios, create a REST service to perform the same calculations.

Data Mapping

Mapping data now only supports the Data Mapper grammar. When you import your per-version 5.x projects, any JavaScript grammar mappings for tasks that are still supported in 5.x are automatically converted to Data Mapper mapping grammar.
After migration, problem markers are generated if you have any scripted mappings that do not have explicit return statements. To fix this, then you must explicitly return the appropriate value to map to the target data at the end of the scripted mapping.

Participant Configurations

As incoming message activities and WSDL invocation are not supported, you should remove your web service provider participants.

Secondly, for service tasks that invoke REST services, the participant configuration has changed. On import, the SOAP HTTP consumer participants are converted to REST Service Consumer Participants and the shared resource instance name is preserved. This means that in the 5.x system BPM Administration you can replicate the HTTP Client shared resource instances found in your existing 4.x system and replace the end-point information with the details REST service that is the equivalent of the former SOAP HTTP service. The shared resource instance name therefore matches the preserved name in the design-time Studio projects.

Multi-Instance Sub-Processes

There is a change in the processing of multi-instance sub-processes.

In multi-instance sub-process call tasks instances, the individual sub-process can complete in any order. Output mappings can be made from a non-array sub-process parameter into an array field in the calling process. The target array field is populated from the values returned by the sub-process instances.

In pre-version 5.x ActiveMatrix BPM, regardless of the order in which the sub-process instances completed, the output value was assigned into the target array element *in the order in which the sub-process instances were started*. In other words, the return value from the first sub-process instance was always assigned to the first target array field element, the second sub-process instance output value was always assigned to the second target array field element. If the sub-process instances completed in a different order than they were started, then the target field had null values assigned to their corresponding target array field elements.

In version 5.x+ this is no longer the case. The target array field elements are now assigned in the order in which the sub-process instances complete. In other words, if the third sub-process instance completes the first then its output value is the first element in the target array field.

Therefore, for single to multi-instance output mappings, the target list must be configured as **Append to Target List**. To enforce this, a problem marker is raised on any multi-instance sub-process single to multi-instance output data mapping, unless you configure it as **Append to Target List**.

If your version 5.x BPM application is designed to rely on the list element index corresponding to the specific order in which the sub-process instance was invoked, this no longer works in version 5.x onwards.

Scripting

There are some changes to scripting in version 5.x, for example, process data fields are now wrapped in a `data` object and factory methods have changed. Some differences are automatically fixed for you when you import your projects for the first time. Depending on the complexity and nature of your scripts, you might need to fix some scripts manually by addressing the problems markers raised on those scripts.

Retrieving and Setting Process Relevant Data

At run-time, instances of BOM classes are created to represent particular instances of the generic BOM class. These instances are referred to as Business Objects. You can retrieve and set business object attributes, using business data scripting. For example, in pre-version 5.x, you could have the following:

```
var custName = customerInstance.name;
```

and you could set the value of the `Name` business object property as follows:

```
customerInstance.name = "Clint Hill";
```

In version 5.x, process relevant data are now wrapped in a `data` object. To set the same value of the `Name` process parameter or data field property, you must now do the following:

```
data.customerInstance.name = "Clint Hill";
```

Business Object Creation by Factory

> **ⓘ** **Note:** This applies to server-side scripting only. Factory usage in client-side scripting in pre-version 5.x is the same as version 5.x.

Business Objects are created by factories. There is a factory method for each class within the BOM (for example, `createCustomer`, `createOrder`, and so on). For example, in pre-version 5.x, you could have the following:

```
com_example_ordermodel_Factory.createOrder()
```

In version 5.x all BOM factory methods are accessed via a predefined variable named factory. To use the same factory method in version 5.x, you must now do the following:

```
factory.com_example_ordermodel.createOrder()
```

The Arrays for multi-instance child attributes definitely are created and handled automatically so they do not need to be created manually in the script. This means that you cannot assign an array directly. To clear an array you must call the `data.array.length=0;` method. To replace the content of an array you must use `data.array.length=0;` and then use `array.pushAll(sourceArray)` to copy the required data to the array.

> **Note:** You can assign the values of two BOM attributes to the same BOM class instance object. In this situation, changes are made by reference, and the changes are reflected in both places. However, once the script is complete, the data are saved and that single value is stored and, thereafter, behaves as two independent objects. This differs from the behavior in pre-version 5.x, where a business object instance can only be contained by a single container at any given instant.

Working with Enumerated Types (ENUM)

> **Note:** This applies to server-side scripting only. In client-side scripting, enumerated types are handled in version 5.x in the same way they were handled in pre-version 5.x.

If you want to categorize objects as different types, instead of using a number or a free format string, you can use an Enumerated Type (ENUM). In pre-version 5.x, you could use unqualified names to define ENUMs in scripts. In version 5.x, this is no longer supported. You must use a fully qualified name (qualified by the package name) for the enumerations in the script. The qualified name of enumerations to be used in scripts is similar to the factory names, with the qualified name formatted to replace dot '.' by '_' an underscore character.

Secondly, the fully qualified name must now be wrapped in a predefined variable named `pkg` object. For example:

```
com_example_shared_ColorEnum.RED
```

becomes

```
pkg.com_example_shared_ColorEnum.RED
```

Case Data Access Methods

Prior to ActiveMatrix BPM version 5.x, specific JavaScript classes were dynamically generated to handle specific BOM case classes. ActiveMatrix BPM version 5.x has been changed to provide a statically defined (that is, it is always there) JavaScript class that is used for all case classes, where the case class is identified in the method parameters using its fully qualified name (that is, "<bom_package_id>.<case_class_name>".

Upon migration from pre-version 5.0, scripts are automatically changed to use the equivalent new method, assuming there is an equivalent method, and that the referenced Business Data projects have been imported.

Script Utility

The former ScriptUtil JavaScript class has been removed.

- The former `ScriptUtil.copy(...)` methods are automatically replaced with the equivalent `bpm.scriptUtil.copy(...)` method during import migration from BPM 4.x to 5.x.

- Due to the data format change from XML to JSON, the former ScriptUtil.toXML(...) and from ScriptUtil.fromXML(...) methods are no longer available. You can use the new JSON serialization methods instead. For more information, see JSON Serialization Methods .

- The other methods formerly provided by ScriptUtil (for example, date manipulation methods) are no longer available because the equivalents are supported by the native Javascript data types now used in BPM 5.x.

JSON Serialization Methods

The following `bpm.scriptUtil` methods allow you to serialize business objects into or deserialize them from their JSON string representation.

Business data field values to JSON string

```
var jsonString = bpm.scriptUtil.stringify(customer);
```

Returns the JSON representation of the customer object.

JSON string to Business data field values

```
data.Customer= bpm.scriptUtil.parseData(customerJSONString,
"com.xyz.insurance.CustomerClass");
```

- `jsonAsString`: Any valid JSON string that represents the object qualified by the fully qualified name passed

- `"com.xyz.insurance.CustomerClass"`: Fully qualified Business data class name

DateTime Utility

The following utility function allows to create date, time and date-time objects of the types used in pre-version 5.x.

DateTime to DateTime TZ type

```
var date = new Date();
bpm.dateTimeUtil.convertDateTimeToDateTimeTZ(date);
```

The above method converts passed in date object to DateTimeTZ in UTC.

DateTime string to DateTimeTZ type

```
var dateString = data.dateStringParameter;
data.readInDateTimeTz = bpm.dateTimeUtil.convertDateTimeToDateTimeTZ
(dateString);
```

The above method converts date string to DateTimeTZ in UTC.

addDuration() Method

Durations defined as an offset specified in either years, months, days, hours, seconds or milliseconds or some combination of these parameters. These are best defined as separate integer objects for the amount(s) of each standard duration that is required. These can then be added to the part of the base dateTime to give the required end date. This Utility function returns a new date that is the result of adding the duration to the date passed in the argument.

> **i** **Note:** It does not alter the passed in date but returns a new date with duration applied.

Syntax

```
var dateTime=data.myDate;
bpm.dateTimeUtil.addDuration(dateTime, duration);
```

Example 1

To add 2 hours onto a Datetime

```
data.targetDate = bpm.dateTimeUtil.addDuration(dateTime, "PT2H");
```

Example 2

To calculate a datetime corresponding to 36 hours into the future.

```
data.targetDate = bpm.dateTimeUtil.addDuration(dateTime,true,
0,0,0,36,0,0,0);
```

Example 3

To calculate a datetime corresponding to 36 hours ago.

```
data.targetDate = bpm.dateTimeUtil.addDuration(dateTime,false,
0,0,0,36,0,0,0);
```

Regular Expression Patterns for Text Fields No Longer Supported

In pre-version 5.x, when defining a Primitive Type, text fields could be constrained to match certain patterns. Regular expression patterns are no longer supported.

JavaScript or JSON Arrays

In pre-version 5.x a process data field specified as multi-valued data could refer to multiple instances of a business object. Similarly, a BOM class attribute could refer to multiple instances of a business object. Multiple instances of business objects used to be handled using a List object. List objects are no longer supported. In version 5.x, you must use JavaScript or JSON Arrays instead.

DataUtil

The DataUtil utility is no longer supported.
DataUtil. DataUtil provided a single method that allowed you to create a List object for

use in scripting in pre-version 5.x.

For more information, see the JavaScript or JSON Arrays section.

WorkManagerFactory and Process Classes

WorkManagerFactory and Process classes have the following changes:

- WorkManagerFactory is renamed as workManager. WorkManagerFactory JavaScript class has been replaced by bpm.workManagerFactory.

- process JavaScript class has been replaced by bpm.process.

Example 1

```
var theOfferSet = WorkManagerFactory.getWorkItem().getWorkItemOffers();
```

becomes

```
var theOfferSet = bpm.workManager.getWorkItem().getWorkItemOffers();
```

Example 2

```
var myActivityLoopIndex = Process.getActivityLoopIndex();
```

becomes

```
var myActivityLoopIndex = bpm.process.getActivityLoopIndex();
```

Forms

There are some changes to forms in version 5.xActiveMatrix BPM, for example, there are some changes to `get` and `set` methods. Unless stated otherwise, when importing a pre-version 5.x project into version 5.x TIBCO Business Studio - BPM Edition that uses scripts that contain any of the following features, problem markers are generated that you must fix.

There are also changes to business object model data types and scripting that impact forms, for example changes to business object model data types. See Business Object Model Data Types and Scripting.

Scripting API changes

- Script actions, computation actions and validation scripts are all migrated automatically. External JavaScript files must be migrated manually.

- There are some changes to `get` and `set` methods. Previously, properties used to be read or written using a `get` or `set` method respectively. Now, such `get`/`set` method pairs have been replaced by direct properties, which can be used as `lvalues` (to write them) or `rvalues` (to read them). Read-only properties never had a set method, and such properties cannot be used as lvalues. For a full list of the new properties and remaining methods, see "API for Scripting" in the *TIBCO Business Studio™ - BPM Edition Forms User Guide*. `<obj-expr>.getXxx()` and `<obj-expr>.setXxx(<value-expr>)` calls are automatically migrated to use property access syntax `<obj-expr>.xxx` and `<obj-expr>.xxx = <value-expr>` respectively.

- Multi-valued structured-type properties are now JavaScript `Array` objects, not `List` objects. Most List method calls are automatically migrated to use the corresponding Array API. See List.iterator() below for exceptions to this.

- You can set the `array.length` to zero and then use the `array.pushAll(...)` method to copy the contents from another array as required

- In pre-version 5.x, field values could be accessed using `f.controlName`. In version 5.x, `f.xxx` and `f[<expr>]` are replaced by `control.xxx.value` and `control[<expr>].value`, respectively.

- In pre-version 5.x parameter values could be accessed using `p.paramName`. In version 5.x, `p.xxx` and `p[<expr>]` are replaced by `data.[expr]` and `data[xxx]`, respectively.

CSS Best Practice

Version 5.x ActiveMatrix BPM now enforces CSS best practice, by removing all the deprecated configuration properties that used to control rendering of form components by non-CSS means. Following import of a pre-version 5.x project, any such properties (for example, font size, weight, color, and so on,) must be re-applied using pure CSS techniques. As part of this, Font and Layout property tabs and their associated property sections have been removed.

Resource/Model Name Length

You can no longer use long names for your resource/models. If you have used long names, when importing to version 5.x, you might receive problem markers that you must fix.

Duration Forms Component Type No Longer Supported

In pre-version 5.x ActiveMatrix BPM, the duration control allowed the users to specify a duration using a configurable set of temporal units. The duration control is no longer supported in version 5.x.

Custom Controls API changes

The API for custom controls has changed to use direct property access instead of `get/set` method calls. Custom controls supply a JavaScript wrapper script which (in common with all external JavaScript resources) is not migrated. You must migrate the JavaScript wrapper script manually. See "Custom Controls" in the *TIBCO Business Studio™ - BPM Edition Forms User's Guide*.

List.iterator()

For `List.iterator()` and all calls on the resulting Iterator object such as `hasNext()`, `next ()`, `add()`, `remove()` and so on, are not automatically migrated. You must manually refactor any code that uses `List.iterator()`.

Frequently Asked Questions

Frequently asked questions are listed questions and answers that are commonly asked when migrating a 4.x project to 5.x project of TIBCO Business Studio™ - BPM Edition.

Question	Answer
Why can't I import all of my projects at once and instead have to import in the order of dependency hierarchy?	If you import projects in dependency order and resolve most of the migration-related problems in lower-level projects first, then you will find that there are generally fewer problems to deal with in higher-level projects as you will have already addressed the root cause of an issue in the underlying projects first.
Why can't I find the "Distributed Application Archive (DAA) Export" option?	The application development test and production deployment lifecycle are now controlled by the items in the project Deployment context menu. The deployment artifacts for projects are now RASC (Runtime Application Shared Concept) files with a .rasc extension. These deployment artifacts are smaller and generally faster to build and deploy. RASC files can simply be drag-dropped onto the BPMe Administration>Deployment Manager>New Deployment to deploy the project. For more information, see the Deploying Applications section of the *TIBCO® BPM Enterprise, Administration Guide*.
Why is business data now stored as JSON instead of XML?	JSON is the contemporary data format for REST services and the BPM 5.x API is now completely REST based. Also, JSON is more compatible with JavaScript and provides more transparent execution semantics in scripting.
Why are incoming web service requests no longer supported?	You now have to use the general-purpose BPM run-time API's instead.
My Business Data class generalizations have been removed, why?	Generalization is no longer supported, however, the attributes formerly inherited from the generalized class are copied into the sub-class automatically during migration from 4.x.

TIBCO Documentation and Support Services

For information about this product, you can read the documentation, contact TIBCO Support, and join TIBCO Community.

How to Access TIBCO Documentation

Documentation for TIBCO products is available on the Product Documentation website, mainly in HTML and PDF formats.

The Product Documentation website is updated frequently and is more current than any other documentation included with the product.

Product-Specific Documentation

The documentation for this product is available on the TIBCO Business Studio™ - BPM Edition Product Documentation page.

How to Contact Support for TIBCO Products

You can contact the Support team in the following ways:

- To access the Support Knowledge Base and getting personalized content about products you are interested in, visit our product Support website.

- To create a Support case, you must have a valid maintenance or support contract with a Cloud Software Group entity. You also need a username and password to log in to the product Support website. If you do not have a username, you can request one by clicking **Register** on the website.

How to Join TIBCO Community

TIBCO Community is the official channel for TIBCO customers, partners, and employee subject matter experts to share and access their collective experience. TIBCO Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from TIBCO products. In addition, users can submit and vote on feature

requests from within the TIBCO Ideas Portal. For a free registration, go to TIBCO Community.

Legal and Third-Party Notices

SOME CLOUD SOFTWARE GROUP, INC. ("CLOUD SG") SOFTWARE AND CLOUD SERVICES EMBED, BUNDLE, OR OTHERWISE INCLUDE OTHER SOFTWARE, INCLUDING OTHER CLOUD SG SOFTWARE (COLLECTIVELY, "INCLUDED SOFTWARE"). USE OF INCLUDED SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED CLOUD SG SOFTWARE AND/OR CLOUD SERVICES. THE INCLUDED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER CLOUD SG SOFTWARE AND/OR CLOUD SERVICES OR FOR ANY OTHER PURPOSE.

USE OF CLOUD SG SOFTWARE AND CLOUD SERVICES IS SUBJECT TO THE TERMS AND CONDITIONS OF AN AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER AGREEMENT WHICH IS DISPLAYED WHEN ACCESSING, DOWNLOADING, OR INSTALLING THE SOFTWARE OR CLOUD SERVICES (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH LICENSE AGREEMENT OR CLICKWRAP END USER AGREEMENT, THE LICENSE(S) LOCATED IN THE "LICENSE" FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE SAME TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of Cloud Software Group, Inc.

TIBCO, the TIBCO logo, the TIBCO O logo, Business Studio, and TIBCO Business Studio are either registered trademarks or trademarks of Cloud Software Group, Inc. in the United States and/or other countries.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only. You acknowledge that all rights to these third party marks are the exclusive property of their respective owners. Please refer to Cloud SG's Third Party Trademark Notices (https://www.cloud.com/legal) for more information.

This document includes fonts that are licensed under the SIL Open Font License, Version 1.1, which is available at: https://scripts.sil.org/OFL

Copyright (c) Paul D. Hunt, with Reserved Font Name Source Sans Pro and Source Code Pro.

Cloud SG software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. See the "readme" file for the availability of a specific version of Cloud SG software on a specific operating system platform.