TIBCO iProcess® Database Plug-in

User Guide

Version 11.9.0
August 2022

Contents

Introduction

What's New

In this release, the EAIDB design-time plug-in is completely re-developed. It has the capability to access the server database at design-time. As a result, the whole UI paradigm changes from one where the onus is on the user to know and input the correct values to one, where the UI knows about the database and stored procedures etc., and provides a list and validation to map to a stored procedure call.

Also, it allows the usage of the new Global Variables feature to reference database and server details. Input mappings allow to use iProcess Expressions, which helps manipulate data before passing it to the stored procedure and eliminates the need to add additional fields to pass parameters. Though you are familiar with EAIDB, the new guide helps you understand how the current version differs from the 11.8.x and earlier versions.

Creating EAI Database Steps

This section describes how to define EAI Database steps in your iProcess procedures. EAI Database steps enable you to integrate your procedure with database stored procedures.

What are Stored Procedures?

Database stored procedures are a common method of performing simple but repetitive tasks on the database (such as adding new employee records).

They exist as database schema objects, which logically group a set of SQL and other PL/SQL programming language statements together to perform a specific task. You can interact with stored procedures by sending case data to them and receiving back data to use in the case.

By using database stored procedures, you can re-use existing business logic in your iProcess procedure and retain the benefits of stored procedures:

- Security - the schema definition remains private to the stored procedure definer.
- Performance - database processing is performed in the database, which benefits from data caches and avoids unnecessary data transfers.
- Integrity - integrity is improved because reusing the code reduces the chances of introducing coding errors.
- Resource usage - stored procedures are stored in the database global memory so they can be reused by multiple sessions.

For more information about database stored procedures, see the database documentation.

What is the Transaction Scope of EAI Database Steps?

The EAI Database step can only be part of the same transaction that is controlled by the iProcess background process. The stored procedure must not contain any logic to commit or rollback changes made by the stored procedure.

This is because the decision to commit or rollback the transaction involving the stored procedure (based on return codes from the plug-in and other components) is made by the iProcess background process. Doing so could lead to a possible invalid database state.

Prerequisites for Using EAI Database Steps

Before using EAI Database steps in your business process, you need to make sure you meet the following prerequisites:

- You must be connected to the database instance where your stored procedures exist.
- You must have installed the client and server EAI Database Plug-ins when you installed the iProcess Workspace (Windows) and iProcess Engine.
- The stored procedures must not contain any logic to commit or rollback changes made by the stored procedure.

Creating an EAI Database Step

To create an EAI Database step in your procedure, you need to perform the following steps:

1. Define Basic EAI Step Information (name, description, step type, deadline, and audit trail information.)
2. Define the Database Stored Procedure to Call.
3. Specify the Parameter Mappings for the Stored Procedure.
4. Optionally, Define a Stored Procedure to Call if the Step is Withdrawn.

When you have done this, the EAI step type is defined as an EAI Database step and the following icon is displayed:  .

Define Basic EAI Step Information

1. Start the TIBCO iProcess Modeler, click the EAI Step tool  and click in the window where you want to place the EAI Step.
2. In the Properties pane, enter the **Name** and **Description** for the step.
3. In the **EAI Step Type** drop-down list, select **EAIDB - Database EAI step plug-in**.

   You must enter this when you first create the step; it cannot be changed later. The list box displays EAI step types that have been installed as client EAI plug-ins.
4. Click the **Deadlines** button if you want to enter deadline information for this step. You can also enter the step duration if you are using case prediction. For more information about defining deadlines and using case prediction, see the “Using Deadlines” topic in the *TIBCO iProcess Modeler Basic Design*.
5. On the **Options** tab, click the **Ignore Case Suspend** check box if you want the step to still be processed as normal while a case is suspended by an iProcess Objects or SAL application.

   If **Ignore Case Suspend** is not checked (the default option), the step is not processed while the case is suspended.

   If you select the **Don’t delete work items on withdraw** option, and the deadline on a work item expires or it is withdrawn as an action (release or deadline expire) of another step:

   — the deadline actions are processed.

   — the step remains outstanding (the step remains in the work queue or the sub-procedure case is not purged).

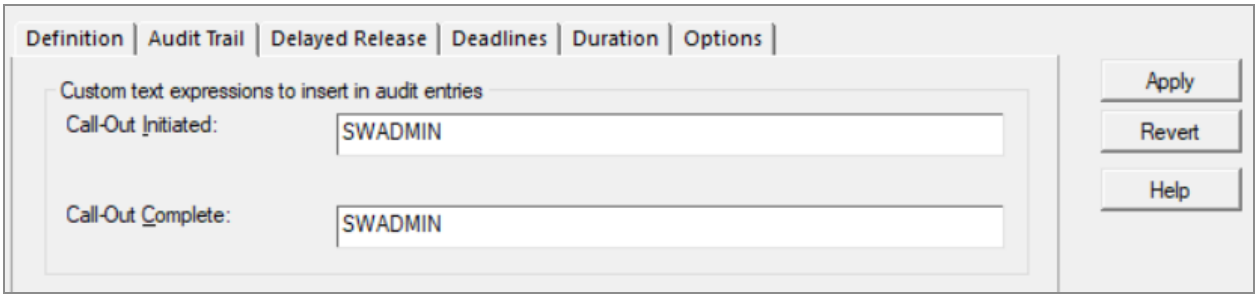
   — when the step is withdrawn, DBMS stored procedure configured in the **Withdraw Step Stored Procedure** tab is not processed.

   — when the step is released (or the sub-procedure case completes) the normal release actions are not processed but the case field data associated with the release step (e.g. the field values set in a normal step whilst in a work queue or the output parameters of a sub-case) is applied to the main case data.

For more information about the other check boxes on the **Options** tab, see the "Defining Call Definition Options" topic in *TIBCO iProcess® Modeler Advanced Design* guide.

Note: Cases can only be suspended and re-activated from an iProcess™ Objects or SAL application. Audit trail messages indicate whether a case is active or suspended. For more information about suspending cases, see the iProcess Objects documentation.

6. Click the **Audit Trail** tab to define custom audit trail entry expressions. You can define text expressions that are evaluated when the step is processed and inserted as the %USER value in the audit trail entries.

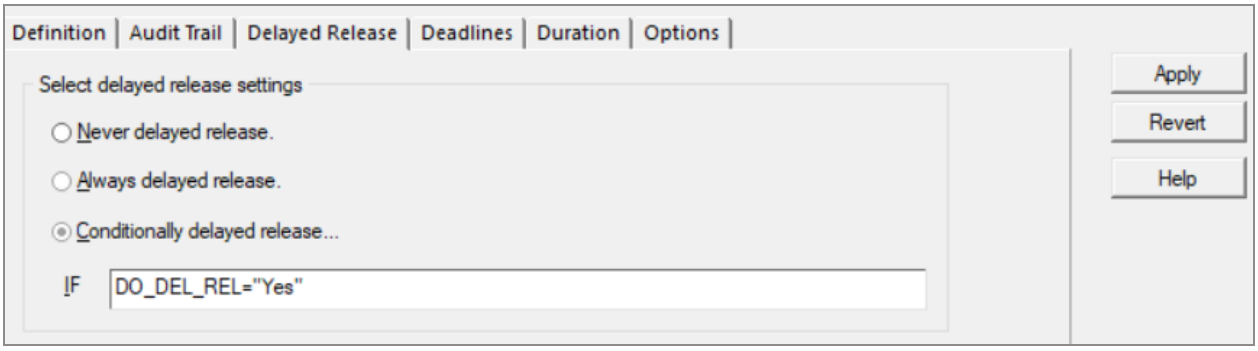


You must enter a value in both fields or leave them both empty.

— In the **Call-Out Initiated** field, enter a valid text expression that replaces the %USER value in the audit trail when the call out is initiated.

— In the **Call-Out Complete** field, enter a valid text expression that replaces the %USER value in the audit trail when the call out is complete.

7. Click the **Delayed Release** tab to define the delayed release setting.



Select one of the following options:

— **Never delayed release** - The step is never set to delayed release so it is released immediately.

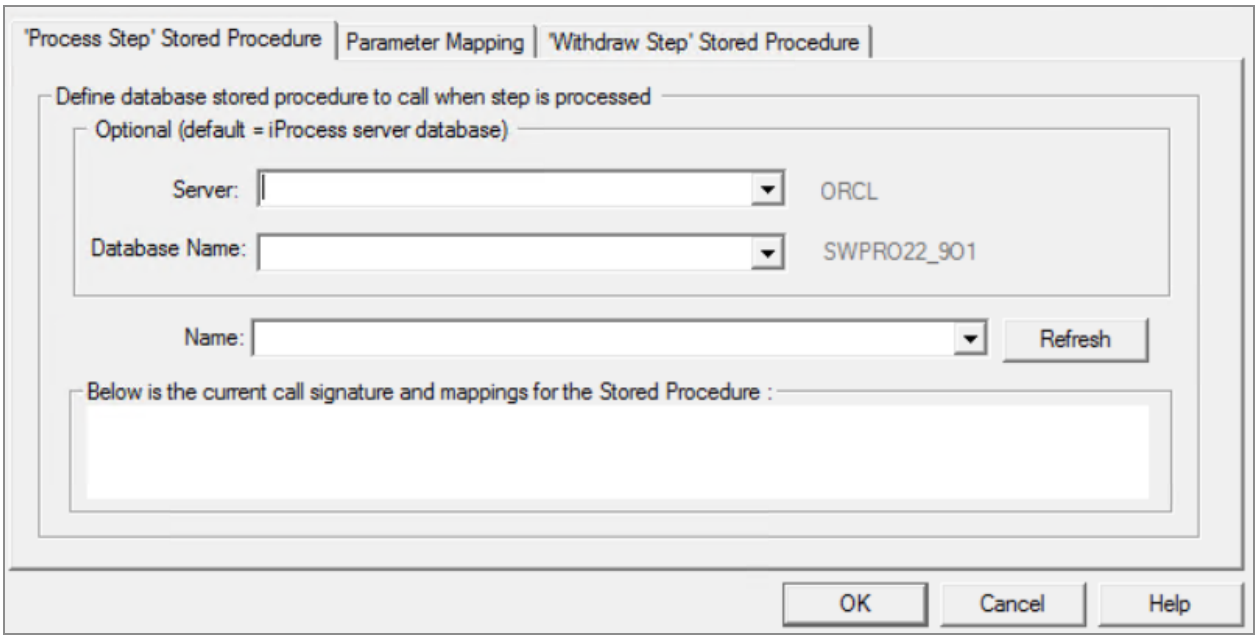
- **Always delayed release** - The step is released by the external application.
- **Conditionally delayed release** - The step is delayed release if a specific condition expression evaluates to True. If you select this option, you need to enter a valid condition expression following the IF statement. For example, IF DO_DEL_REL= “Yes”.

Note: The run-time plug-in on the server can override these settings if it returns a status of Delayed Release when the step is processed.

8. Click the **Definition** tab, then click **Edit**. The **Database EAI Step** dialog box is displayed with the **'Process Step' Stored Procedure** tab highlighted.

Define the Database Stored Procedure to Call

On the **'Process Step' Stored Procedure** tab, you can define the Database stored procedure that you want to call from this step.



1. In the **Server** drop-down list, enter a server name or select one from the list of available servers.
2. In the **Database Name** drop-down list, enter a database name or select one from the list of available databases.

Note:
- The **Server** and **Database Name** drop-down lists display the list of global variables with "$" as prefix.
- If you do not enter the server and database names, they would default to iProcess Engine's database and server names.
- A label is displayed to the right of both the **Server** and **Database Name** drop-down lists, which populates with default and with the selected global variable.

If the stored procedure is not in the local database instance, you must provide information as to its location:

a. For SQL Server, specifying the **Server** name and **Database Name** is optional, where **Server** is the SQL server instance (e.g. machine name) and **Database Name** is the name of the database within that instance.

b. For Oracle, specifying the **Server** name and **Database Name** is optional. Server (previously called Database Link Name) describes the Oracle database link that enables remote access to the Database Name (previously called Schema) which describes the Oracle Schema where the package and/or stored procedure live.

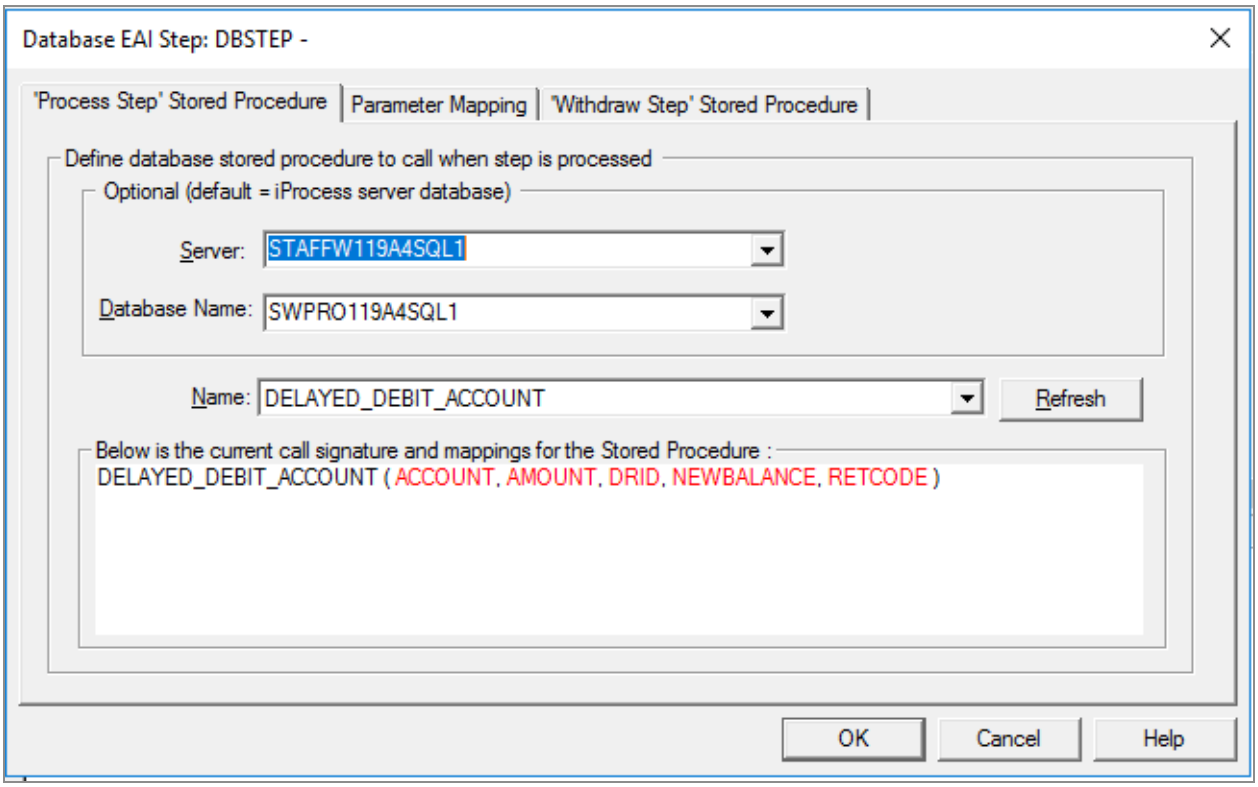
Note:
If you do not specify the **Server** and **Database Names** for SQL server or Oracle, the system fetches the procedures from the iProcess Engine's database.

c. For DB2, the server and database names that are grayed out are not supported. The server and database names are fetched from the local database instance.

3. Click the **Refresh** button next to the **Name** drop-down list.

Note: After clicking the **Refresh** button, filter is applied on the **Server** and **Database Name** drop-down lists and based on that filter, stored procedures are fetched into the Name drop-down.

4. In the **Name** drop-down list, you can view a list of stored procedures.



Warning: The following characters are invalid and must not be used in the name of the DBMS stored procedure:

' ', '\t', '\n', '^', '?', '.', null (ascii code 0), ':', '@'

The full stop is permitted just once and is assumed to imply the separation of procedure owner from procedure name.

Note: If you enter an incorrect server or database name, an error message "**The stored procedure for the selected server and database does not exist**" is displayed on the screen.

5. In the **Name** field, select a stored procedure.

   The text box at the bottom of the dialog box displays any current parameter mappings that are defined in the **Parameter Mapping** tab.

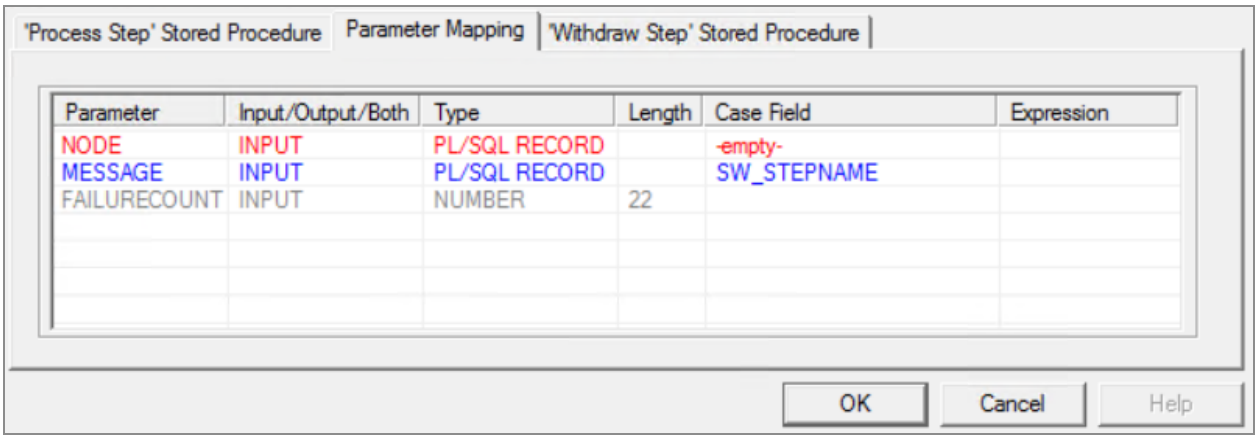
Note: Initially, all the parameters are displayed in Red or Grey as they are not mapped. Go to the **Parameter Mapping** tab to complete the mapping and then, you can revisit this screen and see the complete signature.

The arguments are displayed in three colors based on the parameter mapping.

Red - when a **Required** parameter is not mapped

Blue - when the parameter is mapped

Grey - when an **Optional** parameter is not mapped



After entering all the details on the **'Process Step' Stored Procedure** tab, map the parameters in the **Parameter Mappings** tab.

Note: If you click **OK** without navigating to the **Parameter Mappings** tab, an error message "**You have not mapped the parameters yet**" is displayed, and the EAI step displays as incomplete (work in progress icon).

Restrict Access to Stored Procedures

You can restrict the user access to stored procedures by setting a filter to avoid unauthorized access. In the **Name** drop-down, you can filter the stored procedures. Once the filter is set, you can view the stored procedures relevant only to the you. You can import and export the EAI settings.

To filter the stored procedures in the **Name** drop-down , you can import the filter settings by using the following command:

```
swadm IMPORT_EAI_SETTING <EAISTEP_TYPE> <Complete filename with path>
```

The IMPORT_EAI_SETTING filter can have P+ and P- commands, which filter in a sequential order by creating subsets in each step execution.

For example, there are the following stored procedures in the database SP_GET_PROC1, SP_GET_PROC2, SP_GET_PROC3, SP_SET_PROC1, SP_SET_PROC2, and SP_SET_DESCRIPTION.

If you require a list of all the stored procedures present in the database, you must set the following filter:

```
P+*
```

where:

- P is a procedure.
- + appends and fetches the procedures (while '-' excludes the procedures)

Note: '+' is applied on the main set of the stored procedures while '-' is always applied on the resultant set obtained after filtering the stored procedures.

- * is a wild character that matches zero or more characters.

Note: ? is another wild character that matches only a single character.

If you set the following filter on the stored procedures, observe how the filter is applied in each step execution and what stored procedures are obtained at the end.

```
P+*
P-SP_SET*
P+SP_SET_DESCRIPTION*
```

- In the first step, P+* fetches all the stored procedures from the database.

  ```
  SP_GET_PROC1
  SP_GET_PROC2
  SP_GET_PROC3
  SP_SET_PROC1
  SP_SET_PROC2
  SP_SET_DESCRIPTION
  ```

- In the second step, the filter P-SP_SET* excludes all the stored procedures starting with SP_SET i.e., SP_SET_PROC1 and SP_SET_PROC2 from the total list of stored procedures. The following are the stored procedures fetched from the database after you apply the second filter.

```
SP_GET_PROC1
SP_GET_PROC2
SP_GET_PROC3
```

- After applying the final filter P+SP_SET_DESCRIPTION*, the following stored procedures are fetched.

```
SP_GET_PROC1
SP_GET_PROC2
SP_GET_PROC3
SP_SET_DESCRIPTION
```

After importing the EAI setting, you can list the filtered stored procedures in the **Name** drop-down list by clicking the **Refresh** button.

Note: If you have imported a text file for the EAI setting, that text file contains rules for the filters. The stored procedures are fetched from the database and are filtered based on the filters in the text file.

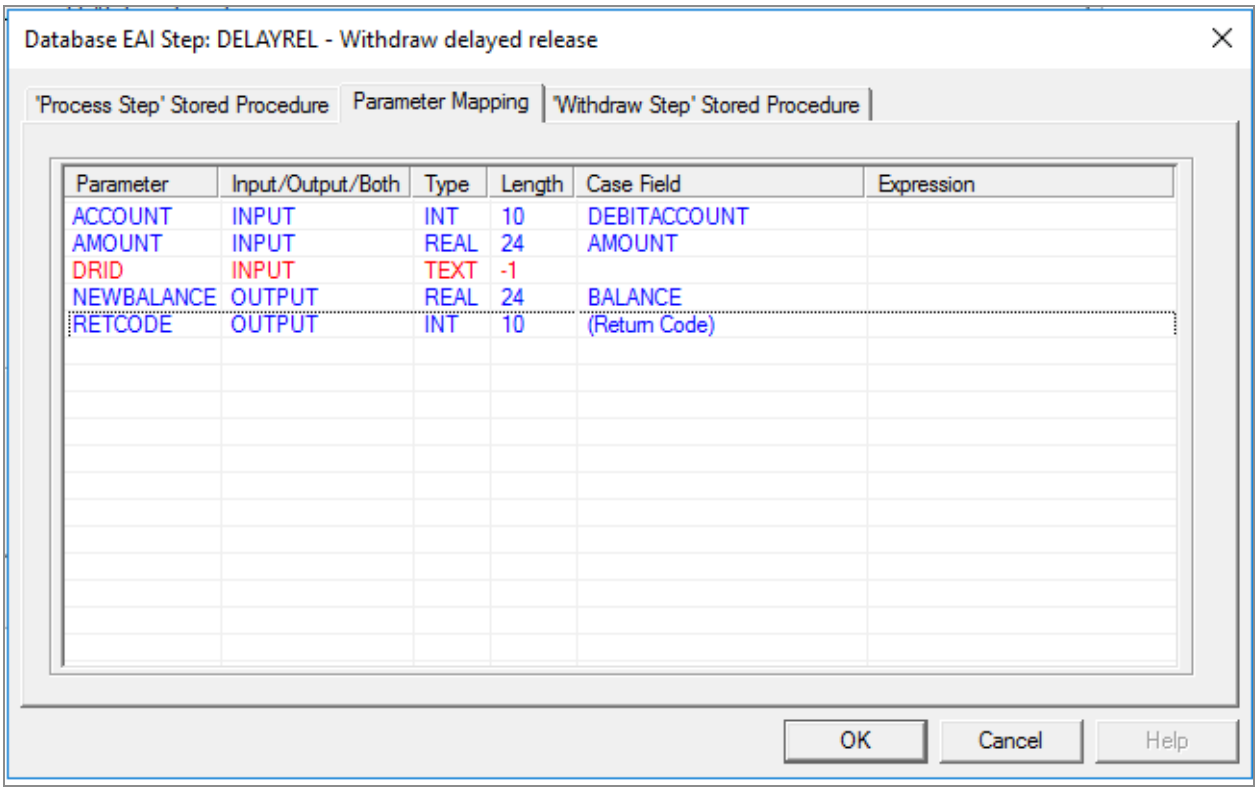
To export the EAI setting, use the following command:

```
swadm EXPORT_EAI_SETTING <EAISTEP_TYPE> [<Complete filename with path>]
```

The export command is a server command, and you must log in to the server to run this command.

Specify the Parameter Mappings for the Stored Procedure

On the **Parameter Mapping** tab, the mappings display a list of all the parameters for the chosen stored procedure. You must map the iProcess case field or an expression to the required parameter respectively.



The **Parameter Mapping** tab has the following columns:

- **Parameter** - displays a declared parameter from the stored procedure that you want to map to an iProcess case field. The example above shows that **ACCOUNT** is a valid parameter used in the stored procedure (**bankapp.debit_account** defined in the **Process Step Stored Procedure** tab)
- **Input/Output/Both** - displays if the parameter is an input parameter, output parameter, or both
- **Type** - displays the parameter type
- **Length** - displays the length of the parameter
- **Case Field** - a drop-down list with iProcess case fields to map to the parameter
- **Expression** - expression to map to the parameter

To map a case field or an expression to a parameter, perform the following steps:

1. From the **Case Field** drop-down list, select the iProcess case field that you want to map the stored procedure parameter to. In the example above, the **ACCOUNT** parameter in the stored procedure is mapped to the DEBITACCOUNT case field in iProcess.

Note:

- From the drop-down list, if you select **{Expression}**, you can map any iProcess expression in the **Expression** column. You can also hard code a value in the **Expression** column. The expression is assigned to the argument in the respective **Parameter** column. In the example above, the ACCOUNT parameter in the stored procedure is mapped to the DEBITACCOUNT case field in iProcess.
- Expressions are validated during design-time. If you do not specify an expression or specify an incorrect expression, the parameter mapping remains in red color. If the expression is mapped correctly, the parameter mapping is displayed in blue color.
- If this step is set up for delayed release, you need to use the **Delayed Release ID** field so that the ID can be passed to the stored procedure for later use when it needs to execute the step.
- The **Return Code** field enables you to check if the stored procedure has been called successfully. For more information about this field, see Return Code below.
- Both the **Delayed Release ID** and **Return Code** fields can only be mapped once in the EAI step.

2. In the **Expression** column, specify the expression for the parameter.

Note:

- The cells under the **Expression** column are enabled only if you select the **{Expression}** option from the drop-down list in the **Case Field** column.
- The expression returns only 256 chars.
- Expressions are only available to map the INPUT parameters, not the OUTPUT or BOTH as these types require a valid return FIELD.

Once the parameter mapping is completed, the arguments are displayed in blue color.

3. Click **OK** to add your mappings.

If you do not map any argument, the **Must supply mandatory casefields** warning is displayed.

Once the mapping of all parameters is completed, the mapping is also displayed in the stored procedure signature on the **'Process Step' Stored Procedure** tab.

Return Code

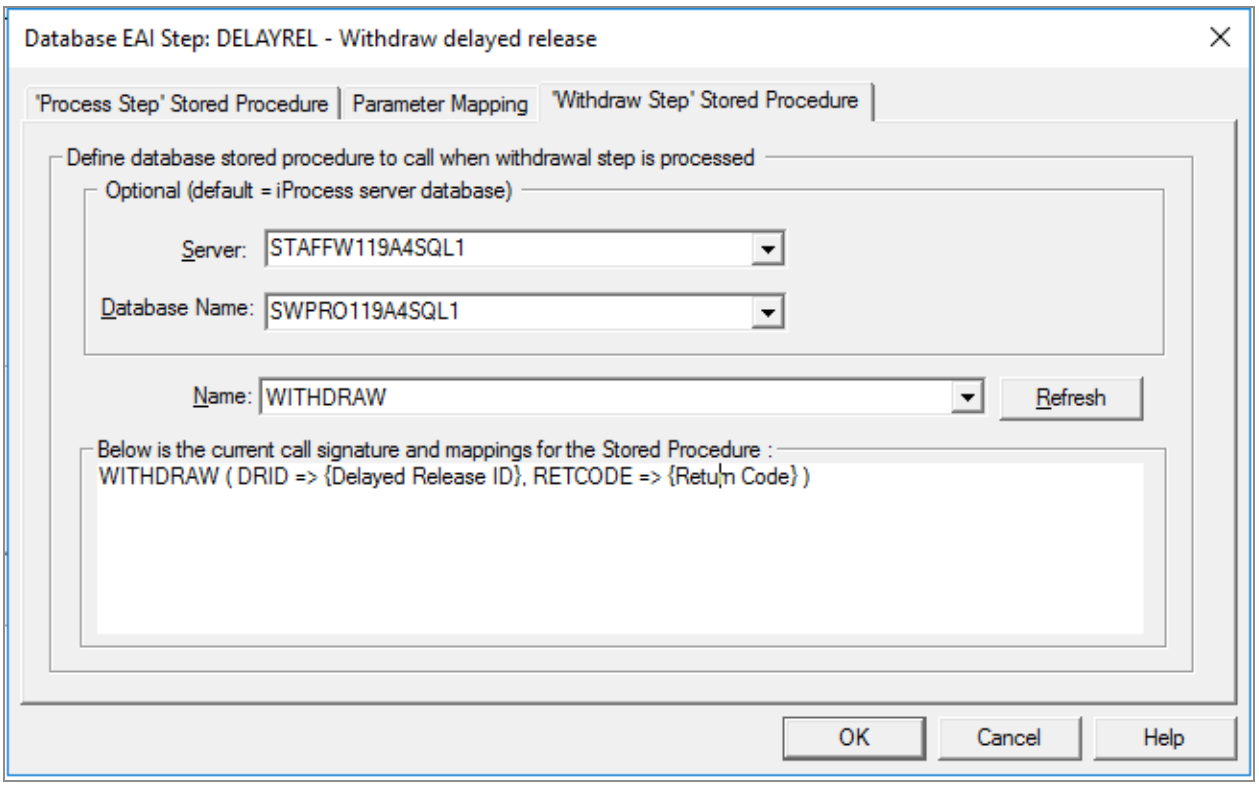
The Return Code field is a special field that, by default, is not visible to any other Task within the Process. The possible return conditions are as follows:

Return Code	Description
1	Indicates successful completion of a stored procedure that is used for the withdrawal of an EAI Database step. If the step is not a withdrawal, a value of 1 is treated as an error (equivalent to –1).
2	The procedure processed completely and its portion of the workflow transaction can be committed if the other steps in the transaction are successful.
3	Indicates successful completion of the stored procedure but treats the step as delayed release. Instead of using the delayed release settings defined for the step, the iProcess background waits for an APPRELEASE command to be issued before the case continues. All return parameters are valid and can be used immediately in other tasks.
–1	Indicates that the stored procedure detected an error. This causes the iProcess background to abort the workflow transaction. The stored procedure executes, but its effects are rolled back.
Other	Error (equivalent to –1).

Define a Stored Procedure to Call if the Step is Withdrawn

On the **Withdraw Step Stored Procedure** tab, you can define a DBMS stored procedure that is called when the EAI step is withdrawn (i.e. on deadline expiry, as an action of another step or on a case purge or close).

Note: This step is optional. Withdrawal of EAI steps only apply to delayed released EAI steps. This means that delayed released has to be selected on the **Delayed Release** tab of the Properties pane.



The stored procedure must be defined so that it takes two parameters. The first needs to be an input parameter, which is passed to the delayed release ID for the step. The second parameter must be an output parameter, which is used for the return status from the stored procedure.

1. In the **Server** drop-down list, enter a server name or select one from the list of available servers.
2. In the **Database Name** drop-down list, enter a database name or select one from the list of available databases.
3. Click the **Refresh** button next to the **Name** drop-down list.
4. From the **Name** drop-down, select a stored procedure. You can see the list of stored procedures with only two arguments.

   The text box at the bottom of the dialog box displays the call signature and parameter mappings with two arguments.

5. Click **OK**.

Define Delayed Release Settings

The **Delayed Release** tab enables you to define the delayed release settings for the step.

You should do this if you set delayed release for the EAI step (by choosing **Always** or **Conditional** on the **Delayed Release** tab of the Properties pane). In this case, the step is not automatically released. The release actions are not processed until the external application sends an APPRELEASE command to iProcess.

Example Procedures Using EAI Database Steps

When you install the iProcess Database Plug-in, you can also install the example iProcess procedures. The following procedures are included to provide a demonstration of how to create EAI Database steps:

- **TRANSFER** procedure

  This is a simple procedure that uses EAI Database steps to perform a fund transfer using DEBIT and CREDIT steps. In the TIBCO iProcess Modeler, click **Procedure** > **Open** and choose the **TRANSFER** procedure.

- **WDRAW** procedure

  This procedure shows how to withdraw a delayed release EAI Database step.

  The `eaibank.sql` file is an example script that creates all the required tables and stored procedures to run the EAI Database plug-in.

If you installed the sample application in the default directory, the files are located in the following directories:

for SQL Server: `SWDIR\examples\eaisql`

for Oracle: `SWDIR\examples\eaiora`

for DB2: `SWDIR\examples\eaidb2`

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 TIBCO Product Documentation website, mainly in HTML and PDF formats.

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

Product-Specific Documentation

The following documentation for this product is available on the TIBCO iProcess® Workspace (Windows) Product Documentation page:

- *TIBCO iProcess® Workspace (Windows) Release Notes*

  Read the release notes for a list of new and changed features. This document also contains lists of known issues and closed issues for this release.

- *TIBCO iProcess® Workspace (Windows) Installation*

  Read this manual for instructions on site preparation and installation.

- *TIBCO iProcess Suite Documentation Library*

  This library contains all the manuals for TIBCO iProcessWorkspace (Windows), TIBCO iProcess® Modeler, and other TIBCO products in TIBCO iProcess Suite. The manuals for TIBCO iProcess® Database Plug-in and TIBCO iProcess® Modeler are the following:

  - *TIBCO iProcess Workspace (Windows) User Guide*
  - *TIBCO iProcess Modeler Getting Started*
  - *TIBCO iProcess Modeler Procedure Management*
  - *TIBCO iProcess Modeler Basic Design*

- *TIBCO iProcess Modeler Advanced Design*
- *TIBCO iProcess Modeler Integration Techniques*
- *TIBCO iProcess Expressions and Functions Reference Guide*
- *TIBCO iProcess Workspace (Windows) Manager's Guide*
- *TIBCO iProcess COM Plug-in User Guide*
- *TIBCO iProcess Database Plug-in User Guide*
- *TIBCO iProcess Email Plug-in User Guide*
- *TIBCO iProcess Script Plug-in User Guide*
- *TIBCO iProcess Plug-in SDK User Guide*

Other TIBCO Product Documentation

When working with TIBCO iProcess® Database Plug-in, you may find it useful to read the documentation of the following TIBCO products:

- TIBCO ActiveMatrix BusinessWorks™
- TIBCO Business Studio™
- TIBCO Enterprise Message Service™
- TIBCO Hawk®
- TIBCO Rendezvous®

How to Contact TIBCO Support

Get an overview of TIBCO Support. You can contact TIBCO Support in the following ways:

- For accessing the Support Knowledge Base and getting personalized content about products you are interested in, visit the TIBCO Support website.
- For creating a Support case, you must have a valid maintenance or support contract with TIBCO. You also need a user name and password to log in to TIBCO Support website. If you do not have a user name, 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