

MiCollab Advanced Messaging TeamQ Data Link Reference

For version 6.1 and above

Notice

The information contained in this document is believed to be accurate in all respects but is not warranted by Mitel Networks™ Corporation (MITEL®). Mitel makes no warranty of any kind with regards to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The information is subject to change without notice and should not be construed in any way as a commitment by Mitel or any of its affiliates or subsidiaries. Mitel and its affiliates and subsidiaries assume no responsibility for any errors or omissions in this document. Revisions of this document or new editions of it may be issued to incorporate such changes.

No part of this document can be reproduced or transmitted in any form or by any means - electronic or mechanical - for any purpose without written permission from Mitel Networks Corporation.

Trademarks

The trademarks, service marks, logos and graphics (collectively "Trademarks") appearing on Mitel's Internet sites or in its publications are registered and unregistered trademarks of Mitel Networks Corporation (MNC) or its subsidiaries (collectively "Mitel") or others. Use of the Trademarks is prohibited without the express consent from Mitel. Please contact our legal department at legal@mitel.com for additional information. For a list of the worldwide Mitel Networks Corporation registered trademarks, please refer to the website: <http://www.mitel.com/trademarks>.

© Copyright 2016, Mitel Networks Corporation

All rights reserved

Contents

Preface	5
References	5
Documentation	5
Documentation Updates	5
Help	6
Document Conventions	6
TeamQ Overview	7
TeamQ in General	7
TeamQ Data Link	7
Configuration File	9
Database Connections	9
Assembly References	10
Data Queries	11
Database Query	12
Database Query Parameters	13
Assembly Query	14
Assembly Query Parameters	15
Input Item Identifiers	17
Input Item Property Names	18
Link Command Strings	19
Call Data Parameter Labels	21
Call Attribute Labels	21
Caller Input Labels	21
Examples	23
Database Query Using SQL Statement	23
Database Query Using Stored Procedure	23
Assembly Query	24
TeamQDataLinkConfig.xml	26

Preface

This guide provides references to TeamQ Data Link.

References

A catalog of technical documentation is included on the MiCollab AM Installation Media. If you are installing any advanced applications, such as Networking and Fax Server applications, you should refer to the appropriate technical documentation for application and installation information.

Documentation

The technical documentation is produced in the PDF format and requires the PDF reader to view it. The documentation set for this MiCollab AM includes the following documents and resources:

- **Developer Resources.** Contains programming guides and API references for developers for integrating the server clients and web applications with MiCollab AM.
- **Integration Technical Notes (ITN).** Contains a set of guides that describe the integration methods and instructions for a variety of phone systems to work with MiCollab AM. The ITNs are generally used by resellers or administrators who are experienced with MiCollab AM and familiar with the integration procedures and terminology.
- **Quick Reference Card (QRC).** Contains shortcuts and quick instructions telling subscribers how to access and use the messaging system.
- **Server Documentation.** Available as a PDF only. Contains administrative guides for administrators about installing, configuring, and administering the messaging system, and user guides for subscribers about accessing the messaging system and checking and sending messages.
- **Spare Parts Documentation.** Contains a set of guides that describe the instructions for installing and configuring hardware parts to work with MiCollab AM. These documents are written for Mitel certified MiCollab AM technicians who are experienced with MiCollab AM and familiar with the procedures and terminology.
- **Software Release Notice (SRN).** This notice introduces the new features, capabilities, and hardware/software requirements for the corresponding MiCollab AM version.

Documentation Updates

Documentation updates may be available from the following sources:

- Mitel certified technicians can view or download documents and program files from our partner web site: connect.mitel.com/connect

Help

The primary source of information about MiCollab AM is the online help available within any of its administrative utilities. You can access **Help** as follows:

- Click the **Help** button in the dialog box or window in which you are working
- Press the **F1** key at any time.

Document Conventions

The following conventions are used in this document:

- **Key Names.** Names of keys on the keyboard are shown in a box.

Example: **Enter**

When two keys must be pressed simultaneously, they are joined by a + sign.

Example: **Alt** + **Tab**

- **Reference to Document.** *Italics* fonts can also signify the titles of other documents.

Example: Refer to *System Installation Guide*.

- **UI Element Names.** Names of UI elements such as dialog windows, screens, menu items, tabs, buttons, icons, etc. are shown in bold.

Example: On the **Startup** screen, click the **Start** icon.

- **User Input.** Information required to be typed or spoken is shown in italics.

Example: Type the password *voicemail*.

- **Warning, Caution, Important, and Notes.** Text for the contents that require attention are shown as follows:

WARNING A warning paragraph advises you of circumstances that can result in the loss of data, harm to the system server platform, or personal harm.

CAUTION Failure to follow these recommendations can result in unauthorized access to the system and consequent loss of data.

IMPORTANT An important paragraph gives decision-making information or informs you of the order in which tasks need to be completed.

NOTE A note gives additional information, provides an explanation, or indicates an exception to the information in the preceding text.

TeamQ Overview

TeamQ in General

TeamQ® is an advanced application available for the MiCollab Advanced Messaging (MiCollab AM) system that works in conjunction with the UConnect™ CEBP (Computer Enabled Business Process) software module.

Using a Windows desktop client and UConnect call-flow script, TeamQ effectively adds call center style capabilities to the MiCollab AM system. Incoming calls can be queued for a team of agents, and the agents can view the calls and make call handling decisions from their desktop.

The TeamQ Data Link can be used by the TeamQ UConnect script to validate caller input and retrieve data from external sources for display to agents in the **TeamQ Desktop** Client. External data sources can include common database servers and custom-written .NET assemblies.

This document contains detailed information related to the TeamQ Data Link functionality, configuration and implementation.

TeamQ Data Link

The TeamQ Data Link can be used by the TeamQ UConnect script to validate caller input and retrieve data from external sources for display to agents in the **TeamQ Desktop** Client.

Data from external sources can be displayed to agents in a cell of a TeamQ call record, and can also be attached to call records as [Link Command Strings](#) that appear in the context-menu for the call. In addition, the priority value assigned to a call can be changed in order to alter the position in queue of a call.

External data sources supported include common database servers and custom-written .NET assemblies (a .NET assembly is also commonly referred to as a "DLL"). In either case, the TeamQ UConnect script uses query parameters to define the data exchange specifics and how the data is presented to TeamQ agents.

Input parameters supply data to the query, and can consist of static text as well as [Call Data Parameter Labels](#) that correspond to caller input or other call-related data. For example, input values may include a caller-entered identification number and the telephone number from which the call originated.

Output parameters receive data from the query, and can consist of text generated by the query as well as [Call Data Parameter Labels](#). Resulting values are assigned to [Input Item Property Names](#) for presentation to agents.

Static parameters can be used for assigning common, fixed values to [Input Item Property Names](#) in order to, for example, supply a common link command that is not derived from query output.

In the TeamQ Data Link, **Database Queries** target back-end database files or servers, and **Assembly Queries** target .NET assemblies.

Database Queries

Database queries enable the TeamQ UConnect script to validate input and retrieve external data by executing a query-language (e.g. T-SQL for SQL Server) query or stored procedure against a database file or server. The query can target Microsoft SQL Server, Oracle, or any database that has a Windows compatible ODBC or OLE DB provider. Other product-specific database providers can be installed and used as long as they are .NET DbProviderFactory compliant (consult the specific database provider documentation).

Configuring the TeamQ Data Link to support a database query involves specifying the details for a specific connection in the [Database Connections](#) section of the configuration file, and then adding a [Database Query](#) that utilizes the connection to the [Data Queries](#) section of the configuration file.

.NET Assembly Queries

Assembly queries enable the TeamQ UConnect script to validate input and retrieve external data by calling a method in an external, custom-developed .NET assembly. Data is exchanged using the arguments of the method and a return value indicating whether the method call was successful.

Configuring the TeamQ Data Link to support an assembly query involves specifying the details for a specific assembly in the [Assembly References](#) section of the configuration file, and then adding an [Assembly Query](#) that utilizes the reference to the [Data Queries](#) section of the configuration file.

Configuration File

The configuration file is an XML file containing the configuration information for the TeamQ Data Link. In order to be processed, the file must be in the same folder as the executing TeamQ UConnect script. By default, this is the **\CX\UConnect\Script** folder.

NOTE When executing UConnect scripts on an UConnect Remote server, the configuration file must be located in the corresponding folder on the UConnect Remote server.

NOTE All XML element names, attribute names, and value constants are case sensitive.

Root XML Element Format

```
<TeamQDataLinkConfig>
  <DatabaseConnections></DatabaseConnections>
  <AssemblyReferences></AssemblyReferences>
  <DataQueries></DataQueries>
</TeamQDataLinkConfig>
```

Table 1. TeamQDataLinkConfig Child Elements

Name	Description
DatabaseConnections	Contains specifications for one or more database connections used in data link queries. Refer to Database Connections .
AssemblyReferences	Contains specifications for one or more .NET assemblies used in data link queries. Refer to Assembly References .
DataQueries	Contains configuration details for one or more data link queries that utilize a database connection or assembly reference. Refer to Data Queries .

Database Connections

The **DatabaseConnections** element consists of one or more **Connection** child elements. Each **Connection** element provides specific details required to establish a database connection. Configured connections are referred to by name in [Database Query](#) configurations.

XML Element Format

```
<DatabaseConnections>
  <Connection name="" connectionString="" providerName="" />
</DatabaseConnections>
```

Table 2. Connection Element Attributes

Name	Description
name	Connection configuration name. This name is referred to in Database Query elements. If blank the connection element will be ignored.
connectionString	Complete database connection string appropriate for specified provider.
providerName	Full name of .NET DbProviderFactory compliant provider.

Connection Data Providers

Connections are supported to .NET DbProviderFactory compliant data providers. Providers other than those listed below will need to be installed separately.

Table 3. .NET Framework Data Providers

Full Name	Description
System.Data.Odbc	.NET Framework Data Provider for ODBC.
System.Data.OleDb	.NET Framework Data Provider for OLE DB.
System.Data.OracleClient	.NET Framework Data Provider for Oracle.
System.Data.SqlClient	.NET Framework Data Provider for SQL Server.

NOTE OLEDB providers and ODBC drivers that are not native to the .NET Framework would need to be installed separately on the platform.

Assembly References

The **AssemblyReferences** element consists of one or more **Assembly** child elements. Each **Assembly** element provides specific details required for the Data Link to identify and load a .NET assembly. Configured assemblies are referred to by name in [Assembly Query](#) configurations.

IMPORTANT Assemblies must be installed into the global assembly cache, or in the same folder as the TeamQ UCCONNECT script or a folder referenced in the environment path.

XML Element Format

```
<AssemblyReferences>
  <Assembly name="" fullname="" namespace="" class="" />
</AssemblyReferences>
```

Table 4. Assembly Element Attributes

Name	Description
name	Assembly file name, without the file extension. This name is referred to in Assembly Query elements. If blank the assembly element will be ignored.
fullname	Optional. Long form full name of the .NET assembly (see below).
namespace	Optional. Namespace within the assembly containing the class. Defaults to name if not specified.
class	Optional. Class within the namespace containing the method(s) to be called by data queries. Defaults to name if not specified.

Long Form Assembly Name

The long form of an assembly name, also known as the display name, consists of the simple name along with the version, culture, public key token, and optionally processor architecture.

For example:

"MyAssembly, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null"

Data Queries

The **DataQueries** element consists of one or more **DataQuery** child elements. Each **DataQuery** element contains query-specific configuration settings including the target TeamQ input item and whether the query will be used to validate caller input. **DataQuery** elements can refer to either database or assembly queries.

XML Element Format

```
<DataQueries>
  <DataQuery id="">
    <DatabaseQuery></DatabaseQuery>
    <AssemblyQuery></AssemblyQuery>
    <ValidateInput>0</ValidateInput>
  </DataQuery>
</DataQueries>
```

Table 5. DataQuery Element Attributes

Name	Description
id	Identifier of the TeamQ input item that the query applies to. If blank the element will be ignored. Refer to Input Item Identifiers .

Table 6. DataQuery Element Child Elements

Name	Description
DatabaseQuery	Element containing database query details. Refer to Database Query .
AssemblyQuery	Element containing assembly query details. Refer to Assembly Query .
ValidateInput	Whether the query is also used to validate caller input. <ul style="list-style-type: none"> • 0 – False; query is not used to validate input. • 1 – True; query is used to validate input

Input Validation

Caller input validation is supported by both database and assembly queries. If a query indicates that the caller input is invalid, the caller will be prompted to enter the input again. Refer to [Database Query](#) and [Assembly Query](#) for specific input validation requirements.

Database Query

The **DatabaseQuery** element contains the information needed to retrieve external data by executing a query-language query or stored procedure against a database file or server.

XML Element Format

```
<DatabaseQuery>
  <ConnectionName></ConnectionName>
  <CommandText></CommandText>
  <CommandType></CommandType>
  <Parameters></Parameters>
</DatabaseQuery>
```

Table 7. DatabaseQuery Element Child Elements

Name	Description
ConnectionName	Name of the connection as defined in the Database Connections section.
CommandText	Text of the command consisting of a query-language query or stored procedure name.
CommandType	Type of the command specified in CommandText . <ul style="list-style-type: none"> • Text – Command is a query-language query. • StoredProcedure – Command is a stored procedure.
Parameters	Parameters for the query. See Database Query Parameters .

Return Data and Input Validation

Query-language queries return a result set containing one (1) record of data (additional records are ignored). When being used for input validation, the input is considered valid if a record is returned and invalid if no records are returned.

Stored procedure queries can return a result set and/or output parameters. When being used for input validation, the stored procedure can supply a return value parameter of 0 to indicate the input is invalid or 1 to indicate the input is valid. If a return value is not supplied, input validation will proceed in the same fashion as query-language queries, i.e. whether a record is returned in a result set.

For both query-language and stored procedure queries the fields in a returned data record can be configured as **column** parameters or can be returned with column names that match the names of TeamQ input item property names.

Database Query Parameters

The **Parameters** element consists of one or more **Parameter** child elements. Each **Parameter** element contains information specific to a parameter applicable to the database query.

XML Element Format

```
<Parameters>
  <Parameter name="" direction="" value="" teamqName="" />
</Parameters>
```

Table 8. DatabaseQuery Parameter Element Attributes

Name	Description
name	Parameter name. May also specify the target TeamQ input item property name.
direction	The direction that data will be passed to/from the query. <ul style="list-style-type: none">• column – Data received from the query in the form of a data column in the result set generated by the query.• input – Data supplied to the query. Applies to stored procedure and query-language queries.• inputoutput – Data supplied to and receive from the query. Applies only to bidirectional stored procedure parameters.• output – Data received from the query. Applies only to stored procedure output parameters.• returnvalue – Data received from the query in the form of a stored procedure return value.• static – Fixed data supplied to the parent DataQuery.
value	Value assigned to the parameter. Applicable to input , inputoutput , and static parameters.
teamqName	TeamQ input item property name. Optional if name specifies the property name.

Parameter Attribute Notes

If the value assigned to the **name** attribute is not or cannot be set to the TeamQ input item property name, the **teamqName** attribute must be set to the property name. This often applies to database queries when specifying query input parameters and result set column names.

With query-languages that utilize placeholders, such as a '?' character, for indicating parameter positioning within the query string, the **Parameter** elements must be included in the proper order and include the **teamqName** attribute.

Parameters with a **direction** attribute value of **column** are optional if the column name as it appears in the result set exactly matches a TeamQ input item property name.

The **direction** attribute value **static** is for supplying a fixed value that is not involved in the query and is passed directly back to the parent **DataQuery** for assignment to an input item property, for example to supply a common link that does not involve any data interaction.

NOTE Data values supplied as arguments to stored procedure calls will all be of the **string** data type

Assembly Query

The **AssemblyQuery** element contains the information needed to retrieve external data by executing a method contained in a .NET assembly.

XML Element Format

```
<AssemblyQuery>
  <AssemblyName></AssemblyName>
  <MethodName></MethodName>
  <Parameters></Parameters>
</AssemblyQuery>
```

Table 9. AssemblyQuery Element Child Elements

Name	Description
AssemblyName	Name of the assembly as defined in the Assembly References section.
MethodName	Name of the method to be executed.
Parameters	Parameters for the method call. See Assembly Query Parameters .

Method Return Values and Input Validation

Assembly query methods (aka function calls) must return a Boolean value; either **True** or **False**.

A return value of **True** indicates to the TeamQ script that the method executed successfully and the output data should be assigned to the item properties as configured. If input validation is active for the query, a return value of **True** indicates that the caller input was valid.

A return value of **False** indicates to the TeamQ script that the method did not execute successfully. In this case, output data will not be assigned to any item properties. If input validation is active for the query, a return value of **False** indicates that the caller input was invalid.

Assembly Query Parameters

The **Parameters** element consists of one or more **Parameter** child elements. Each **Parameter** element contains information specific to a parameter applicable to the assembly method call.

XML Element Format

```
<Parameters>
  <Parameter name="" direction="" value="" teamqName="" />
</Parameters>
```

IMPORTANT When configuring assembly queries, parameters must be specified in the order that is expected by the corresponding method in the assembly.

Table 10. AssemblyQuery Parameter Element Attributes

Name	Description
name	Parameter name. May also specify the target TeamQ input item property name. For assembly queries, the parameter name does not have to match the actual argument name as specified in the assembly method.
direction	The direction that data will be passed to/from the method. <ul style="list-style-type: none">• input – Data supplied to the method as an argument.• inputoutput – Data supplied to and receive from the method using the same argument.• output – Data received from the method in an argument.• static – Fixed data supplied to the parent DataQuery.
value	Value assigned to the parameter. Applicable to input , inputoutput , and static parameters.
teamqName	TeamQ input item property name. Optional if name specifies the property name.

Parameter Attribute Notes

Generally, the **name** attribute identifies the target TeamQ input item property name. For example, an output parameter with a name of **IdentificationLink** will instruct TeamQ to add the output text value as a link to the **Identification** input item.

If the **name** attribute is not or cannot be set to the TeamQ input item property name, the **teamqName** attribute must be set to the property name.

The **direction** attribute value **static** is for supplying a fixed value that is not involved in the query and is passed directly back to the parent **DataQuery** for assignment to an input item property, for example to supply a common link that does not involve any data interaction.

NOTE Data values supplied as arguments to method calls will all be of the *string* data type

Input Item Identifiers

Input item identifiers are specified in **DataQuery** elements to identify the specific input item that is associated with the query.

NOTE Input item identifiers are case sensitive.

Table 11. TeamQ Input Item Identifiers

Identifier	Description
CallbackNumber	Callback number input item.
Identification	Identification input item.
Selection	Selection input item.
User0	User defined input item 0.
User1	User defined input item 1.
User2	User specified item 2.
User3	User specified item 3.
User4	User specified item 4.
User5	User specified item 5.
User6	User specified item 6.
User7	User specified item 7.
User8	User specified item 8.
User9	User specified item 9.

Input Item Property Names

Input item property names identify how query data is attached to calls and subsequently presented to agents. Properties are available for display values, the values that appear to agents in the call grids, and for link command values, specifications for links that agents can view and activate by right-clicking on a column or row in the grids.

Properties can be assigned values by any query, regardless of whether the query is associated with the specific input item. For example, a query targeted at the **Identification** input item could set the **User2Value** item property.

In the case of the **CallPriority** property, the actual priority value for the call is changed, which may affect the call's position in queue.

NOTE Input item property names are case sensitive.

Table 12. TeamQ Input Item Property Names

Property Name	Description
CallbackNumberLink	Link command string for callback number item.
CallbackNumberValue	Display value for callback number item.
CallPriority	Call priority value. Can be used to change the priority value of a call.
IdentificationLink	Link command string for identification item.
IdentificationValue	Display value for identification item.
SelectionLink	Link command string for selection item.
SelectionValue	Display value for selection item.
User0Link	Link command string for user input item 0.
User0Value	Display value for user input item 0.
User1Link	Link command string for user input item 1.
User1Value	Display value for user input item 1.
User2Link	Link command string for user input item 2.
User2Value	Display value for user input item 2.

User3Link	Link command string for user input item 3.
User3Value	Display value for user input item 3.
User4Link	Link command string for user input item 4.
User4Value	Display value for user input item 4.
User5Link	Link command string for user input item 5.
User5Value	Display value for user input item 5.
User6Link	Link command string for user input item 6.
User6Value	Display value for user input item 6.
User7Link	Link command string for user input item 7.
User7Value	Display value for user input item 7.
User8Link	Link command string for user input item 8.
User8Value	Display value for user input item 8.
User9Link	Link command string for user input item 9.
User9Value	Display value for user input item 9.

Link Command Strings

Link command strings are used to attach one or more link commands to an input item. Link commands can be accessed and invoked by agents using the **TeamQ Desktop** Client. Typically, the command is used to display information related to the call using an external program, such as a web browser.

Each link command specifies, at a minimum, the target process or file to be executed when the link is invoked. The target process can be an Internet address, hyperlink, specific executable program with arguments, or a file with a file type that is registered with Windows on the agent workstation.

Link commands can optionally contain a string to be displayed to agents in the context menu, as well as parameters to be supplied to the target. Parameters are only applicable if the link target is an executable program.

Specific caller input values can be included in link command fields using [Call Data Parameter Labels](#).

Table 13. Link Command Fields

Field	Description
FileName	Target process or file to be executed when the link is invoked.
Display	Optional value for display in the context menu for a call.
Arguments	Optional arguments supplied to an executable file.

Only the **FileName** field is required. When additional fields are included, the '|' character is used as a field separator and the fields must be in the order shown in the table above.

Examples:

FileName | Display | Arguments

FileName | Display

FileName | | Arguments

NOTE Whitespace included in the above examples is for readability and is not required.

Assigning Multiple Links

Multiple link commands can be assigned to an input item property by separating each link command with the '^ ^' character sequence. Links are added to the context menu in the order in which they appear in the string.

Example:

LinkCommand1 ^ ^ LinkCommand2

Call Data Parameter Labels

Call data parameter labels are utilized in query definitions as a placeholder for an associated data value. Actual data values are substituted for the labels when the query is executed. All data values are supplied as text strings. In cases where no data is available, a blank value will result.

For example, the definition for a query that accepts the Team ID number as an input parameter would include the **^TeamID** label as the defined parameter value. When the query is executed the actual value of the Team ID for the call will be provided to the query in place of the label.

NOTE Call data parameter labels are case sensitive.

Call Attribute Labels

Call attribute labels represent data values related to the call that are not caller input.

Table 14. Call Attribute Parameter Labels

Label	Description
^CalledParty	Called party (DNIS)
^CallingParty	Calling party (ANI, extension, mailbox)
^QueueID	Queue ID
^Priority	Call priority
^TeamID	Team ID

Caller Input Labels

Caller input labels represent input digits provided by callers or assigned to an input item value by a query.

Table 15. Caller Input Parameter Labels

Label	Description
^CallbackNumber	Callback number (defaults to calling party)
^Identification	Identification number
^Selection	Selection input or substitution value

^User0	User input 0
^User1	User input 1
^User2	User input 2
^User3	User input 3
^User4	User input 4
^User5	User input 5
^User6	User input 6
^User7	User input 7
^User8	User input 8
^User9	User input 9

Examples

Database Query Using SQL Statement

The following example assigns a database query-language query to the **Identification** input item. The query is executed against the database named **MyDB** on server **MySqlServer**.

The database query is specified directly, and one input parameter is used to supply the caller-entered identification number to the query.

If the identification number is found in the database table, the query returns a data row containing a column named **CustName**. The value returned in this column is assigned to the **User2** input item. Because a data row is returned the caller input was valid. If the identification number is not found the query does not return a data row, indicating that the caller input was not valid.

```
<DatabaseConnections>
  <Connection name="MyConnectionString"
    connectionString="Data Source=MySqlServer;Initial Catalog=MyDB;Integrated Security=True"
    providerName="System.Data.SqlClient" />
</DatabaseConnections>

<DataQueries>
  <DataQuery id="Identification">
    <DatabaseQuery>
      <ConnectionName>MyConnectionString</ConnectionName>
      <CommandText>Select CustName From MyTable Where CustID = @ID</CommandText>
      <CommandType>Text</CommandType>
      <Parameters>
        <Parameter name="@ID" direction="input" value="^[Identification]" />
        <Parameter name="CustName" direction="column" teamqName="User2Value" />
      </Parameters>
    </DatabaseQuery>
    <ValidateInput>1</ValidateInput>
  </DataQuery>
</DataQueries>
```

Database Query Using Stored Procedure

The following example assigns a database stored procedure query to the **Identification** input item. The name of the stored procedure is **usp_LookupAccountName**, which is contained in a database named **MyDB** on server **MySqlServer**.

The **usp_LookupAccountName** stored procedure has four parameters:

- 1 A return value that indicates whether the caller input is valid.
- 2 An input parameter that receives the caller-entered identification number.

- 3 An output parameter that is used to supply a value for the **User2** input item.
- 4 An output parameter that is used to supply a link command string for the **User2** input item.

The stored procedure will set the return value to 1 if the input is valid and a value of 0 if the input is not valid.

```
<DatabaseConnections>
  <Connection name="MyConnectionString"
    connectionString="Data Source=MySqlServer;Initial Catalog=MyDB;Integrated Security=True"
    providerName="System.Data.SqlClient" />
</DatabaseConnections>

<DataQueries>
  <DataQuery id="Identification">
    <DatabaseQuery>
      <ConnectionName>MyConnectionString</ConnectionName>
      <CommandText>usp_LookupAccountName</CommandText>
      <CommandType>StoredProcedure</CommandType>
      <Parameters>
        <Parameter name="@retval" direction="returnvalue"/>
        <Parameter name="@AccountNum" direction="input" value="^Identification"/>
        <Parameter name="@User2Value" direction="output" teamqName="User2Value"/>
        <Parameter name="@User2Link" direction="output" teamqName="User2Link"/>
      </Parameters>
    </DatabaseQuery>
    <ValidateInput>1</ValidateInput>
  </DataQuery>
</DataQueries>
```

Assembly Query

The following example assigns an assembly query to the **Identification** input item. The name of the method to be executed is **LookupIdentification**. The method is contained in a class named **MyDataQueriesClass** within a .NET assembly named **MyDataQueries**.

The **LookupIdentification** method has four arguments:

- 1 An input parameter that receives the Queue ID number for the call.
- 2 An input parameter that receives the caller-entered identification number.
- 3 An output parameter that the method uses to supply a value for the **User2** input item.
- 4 An output parameter that the method uses to supply a value for the **User3** input item.

The method will return a value of **True** if the input is valid and a value of **False** if the input is not valid.

```
<AssemblyReferences>
  <Assembly name="MyDataQueries" class="MyDataQueriesClass" />
</AssemblyReferences>
```



```

<DataQueries>
  <DataQuery id="Identification">
    <AssemblyQuery>
      <AssemblyName>MyDataQueries</AssemblyName>
      <MethodName>LookupIdentification</MethodName>
      <Parameters>
        <Parameter name="QueueID" direction="input"
value="^QueueID" />
        <Parameter name="Identification" direction="input"
value="^Identification" />
        <Parameter name="User2Value" direction="output" />
        <Parameter name="User3Value" direction="output" />
      </Parameters>
    </AssemblyQuery>
    <ValidateInput>1</ValidateInput>
  </DataQuery>
</DataQueries>

```

TeamQDataLinkConfig.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<TeamQDataLinkConfig xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xsi:schemaLocation='urn:tq TeamQDataLinkConfig.xsd'>
  <DatabaseConnections>
    <Connection name="" connectionString="" providerName="" />
  </DatabaseConnections>
  <AssemblyReferences>
    <Assembly name="" fullname="" namespace="" class="" />
  </AssemblyReferences>
  <DataQueries>
    <DataQuery id="">
      <AssemblyQuery>
        <AssemblyName></AssemblyName>
        <MethodName></MethodName>
        <Parameters>
          <Parameter name="" direction="" value="" teamqName="" />
        </Parameters>
      </AssemblyQuery>
      <ValidateInput>0</ValidateInput>
    </DataQuery>
    <DataQuery id="">
      <DatabaseQuery>
        <ConnectionName></ConnectionName>
        <CommandText></CommandText>
        <CommandType></CommandType>
        <Parameters>
          <Parameter name="" direction="" value="" teamqName="" />
        </Parameters>
      </DatabaseQuery>
      <ValidateInput>0</ValidateInput>
    </DataQuery>
  </DataQueries>
</TeamQDataLinkConfig>
```