

DB2 Web Adapter for Microsoft SQL Server

Version 2.1

The page features a decorative abstract graphic consisting of several overlapping, curved lines in shades of blue and grey. These lines sweep across the page from the top left towards the bottom right, creating a sense of movement and depth. The bottom of the page is dominated by a large, layered graphic of overlapping semi-transparent shapes in various shades of blue and grey, creating a textured, mountain-like appearance.

Active Technologies, EDA, EDA/SQL, FIDEL, FOCUS, Information Builders, the Information Builders logo, iWay, iWay Software, Parlay, PC/FOCUS, RStat, Table Talk, Web390, WebFOCUS, WebFOCUS Active Technologies, and WebFOCUS Magnify are registered trademarks, and DataMigrator and Hyperstage are trademarks of Information Builders, Inc.

Adobe, the Adobe logo, Acrobat, Adobe Reader, Flash, Adobe Flash Builder, Flex, and PostScript are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Due to the nature of this material, this document refers to numerous hardware and software products by their trademarks. In most, if not all cases, these designations are claimed as trademarks or registered trademarks by their respective companies. It is not this publisher's intent to use any of these names generically. The reader is therefore cautioned to investigate all claimed trademark rights before using any of these names other than to refer to the product described.

Copyright © 2012, by Information Builders, Inc. and iWay Software. All rights reserved. Patent Pending. This manual, or parts thereof, may not be reproduced in any form without the written permission of Information Builders, Inc.

Contents

- 1. Using the DB2 Web Query Adapter for Microsoft SQL Server.....5**
 - Preparing the Microsoft SQL Server Environment.....6
 - Configuring the DB2 Web Query Adapter for Microsoft SQL Server.....6
 - Declaring Connection Attributes6
 - Controlling the Connection Scope.....9
 - Managing Microsoft SQL Server Metadata.....9
 - Creating Synonyms.....10
 - Microsoft SQL Server Data Type Support.....18
 - Enabling National Language Support.....23
 - Support of Read-Only Fields.....24
 - Reporting Against a Microsoft SQL Server Stored Procedure.....25
 - Generating a Synonym for a Stored Procedure.....25
 - Creating a Report Against a Stored Procedure.....29
 - Customizing the Microsoft SQL Server Environment.....31
 - Specifying the Cursor Type31
 - Activating NONBLOCK Mode.....32
 - Obtaining the Number of Rows Updated or Deleted.....33
 - Controlling Transactions.....34
 - Specifying the Transaction Isolation Level.....35
 - Optimization Settings.....36
 - Optimizing Requests.....36
 - Optimizing Requests Containing IF-THEN-ELSE Virtual Fields.....38
 - Optimizing Requests if a Virtual Field Contains Null Values.....42
 - Calling a Microsoft SQL Server Stored Procedure Using SQL Passthru.....43

1 Using the DB2 Web Query Adapter for Microsoft SQL Server

The DB2 Web Query Adapter for Microsoft® SQL Server® allows applications to access Microsoft SQL Server data sources. The adapter converts data or application requests into native Microsoft SQL Server statements and returns optimized answer sets to the requesting program.

Topics:

- ❑ Preparing the Microsoft SQL Server Environment
- ❑ Configuring the DB2 Web Query Adapter for Microsoft SQL Server
- ❑ Managing Microsoft SQL Server Metadata
- ❑ Reporting Against a Microsoft SQL Server Stored Procedure
- ❑ Customizing the Microsoft SQL Server Environment
- ❑ Optimization Settings
- ❑ Calling a Microsoft SQL Server Stored Procedure Using SQL Passthru

Preparing the Microsoft SQL Server Environment

In order to take full advantage of the features available through the DB2 Web Query Adapter for Microsoft SQL Server 2005, we strongly recommend using MS SQL Server 2005 with Service Pack 1 (2005.90.2047) or higher. To determine which version of MS SQL Server you are using, please refer to the following article: <http://support.microsoft.com/kb/321185>.

Configuring the DB2 Web Query Adapter for Microsoft SQL Server

In this section:

Declaring Connection Attributes

Controlling the Connection Scope

Before you configure the adapter, be sure to install the Microsoft SQL Server 2000 or 2005 JDBC Driver to the Java Extensions directory: `/QIBM/UserData/Java400/ext`.

Configuring the adapter consists of specifying connection and authentication information for each of the connections you want to establish.

Declaring Connection Attributes

Reference:

Connection Attributes for Microsoft SQL Server

In order to connect to an Microsoft SQL Server database server, the adapter requires connection and authentication information. You supply this information using the Web Query Metadata option. The connection and authentication information is added to the profile you select: the global server profile (`edasprof.prf`), a user profile (`user.prf`), or a group profile (if supported on your platform).

You can:

You can declare connections to more than one Microsoft SQL Server databases by including multiple CONNECTION_ATTRIBUTES commands. The actual connection to the Microsoft SQL Server takes place when the first query that references the connection is issued. If you issue multiple CONNECTION_ATTRIBUTES commands:

Reference: Connection Attributes for Microsoft SQL Server

The *MS SQL Server* adapter is under the *SQL* group folder.

The following list describes the connection attributes for which you can supply values. To complete the attribute declaration, click the *Configure* button.

Connection name

Logical name used to identify this particular set of connection attributes. The default is CON01.

Server (Windows only)

Name of the machine where Microsoft SQL Server is running. If that machine has more than one instance of Microsoft SQL Server installed, provide the server name and the instance name as follows: server\instance.

The server connection attribute will allow users to choose from the list of MS SQL Servers visible within the local network using the SQL Native Client Enumerator.

Please note that due to limitations of the SQL Native Client Enumerator, local network settings and MS SQL Server settings, it is possible that not all operational servers will be visible. If the name of the server you wish to target does not appear, you can enter it manually.

URL (UNIX, IBM i, and z/OS only)

Enter the location URL for the Microsoft SQL Server data source.

Security

There are three methods by which a user can be authenticated when connecting to a Microsoft SQL Server:

- ❑ **Explicit.** The user ID and password are explicitly specified for each connection and passed to Microsoft SQL Server, at connection time, for authentication as a standard login.

This option requires that SQL Server security be set to SQL Server and Windows (for Windows), or else to SQL Server and UNIX.

- ❑ **Password Passthru.** (*Windows only*) The user ID and password received from the client application are passed to Microsoft SQL Server, at connection time, for authentication as a standard login.

This option requires that SQL Server security be set to: SQL Server and Windows.

- ❑ **Trusted.** The adapter connects to Microsoft SQL Server as an operating system login using the credentials of the operating system user impersonated by the server data access agent.

This option works with either of the SQL Server security settings.

User

Primary authorization ID by which you are known to the data source.

Password

Password associated with the primary authorization ID.

Default Database (Windows only)

Name of the default database for the connection. This value is used when a data object is not qualified with the database name.

This parameter is optional. If not specified, it defaults to the database associated with the authorization ID.

Driver name (UNIX, IBM i, and z/OS only)

Name for the Microsoft JDBC driver.

Select profile

Select a profile from the drop-down menu to indicate the level of profile in which to store the CONNECTION_ATTRIBUTES command. The global profile, edasprof.prf, is the default.

If you wish to create a new profile, either a user profile (*user.prf*) or a group profile if available on your platform (using the appropriate naming convention), choose *New Profile* from the drop-down menu and enter a name in the Profile Name field (The extension is added automatically).

Store the connection attributes in the server profile (edasprof).

Controlling the Connection Scope

How to:

Control the Connection Scope

The AUTODISCONNECT command controls the persistence of connections when using the adapter for each of the connections you want to establish.

Syntax: **How to Control the Connection Scope**

You change this setting by expanding the Domains and Reports folders, then right-clicking a subfolder and choosing *Metadata*. Right-click MS SQL Server 200x and select *Change Settings*. The Change Settings pane opens.

SQLMSS

Indicates the adapter. You can omit this value if you previously issued the SQLENGINE command.

FIN

Disconnects automatically only after the session has been terminated. FIN is the default value.

COMMAND

Disconnects automatically after each request. Depending on how often the event occurs, the AUTODISCONNECT command may result in considerable overhead. Almost all of this overhead is not related to the server. It is related to the operating system and the data source.

Managing Microsoft SQL Server Metadata

In this section:

Creating Synonyms

Microsoft SQL Server Data Type Support

Enabling National Language Support

Support of Read-Only Fields

When the server accesses a data source, it needs to know how to interpret the data stored there. For each data source the server will access, you create a synonym that describes the structure of the data source and the server mapping of the Microsoft SQL Server data types.

Creating Synonyms

How to:

Create a Synonym

Create a Synonym From the Web Query Environment

Reference:

Synonym Creation Parameters for Microsoft SQL Server

Mapping Microsoft SQL Table Comments Into a Synonym

Access File Keywords

Managing Synonyms

Synonyms define unique names (or aliases) for each Microsoft SQL Server table or view that is accessible from the server. Synonyms are useful because they hide the underlying data source location and identity from client applications. They also provide support for extended metadata features of the server, such as virtual fields and additional security mechanisms.

Using synonyms allows an object to be moved or renamed while allowing client applications to continue functioning without modification. The only modification required is a redefinition of the synonym on the server. The result of creating a synonym is a Master File and an Access File, which represent the server metadata.

Note that creating a synonym for a stored procedure is described with reporting against a stored procedure, in [Generating a Synonym for a Stored Procedure](#) on page 25.

Procedure: How to Create a Synonym

To create a synonym, you must have previously configured the adapter. You can create a synonym from the Applications or Adapters pages of the Web Console.

1. From the Web Console menu bar, click *Applications*.

The Applications page opens.

2. Click the *New* button and select *Synonym* from the drop-down menu.

The *Select adapter to configure* or *Select connection to create synonym* pane opens.

3. Click a connection for the configured adapter.

The first of a series of synonym creation panes opens.

4. Enter values for the parameters required by the adapter as described in the synonym creation parameters reference.

5. After entering the parameter values, click *Create Synonym*.

The Status pane indicates that the synonym was created successfully.

The synonym is created and added under the specified application directory.

Note:

- ❑ You can also create a synonym from the Adapters page by right-clicking a configured adapter connection and selecting *Create Synonym*.
- ❑ When creating a synonym, if you choose the Validate check box (where available), the server adjusts special characters and checks for reserved words.

Procedure: How to Create a Synonym From the Web Query Environment

1. Expand the *Domains* folder, then expand a domain.
2. Expand the *Reports* folder, right-click a request subfolder, and choose *Metadata* from the menu.
3. In the left-hand Adapter navigation pane, click the database connection (MS SQL Server 200x) and choose *Create Synonym* from the menu. The first of a series of synonym creation pages opens.
4. Enter values for the parameters required by the adapter.
For information about these parameters, see [Synonym Creation Parameters for Microsoft SQL Server](#) on page 11.
5. After entering parameter values, click *Create Synonym*.
Synonyms are created and added under the specified application directory.
The Status pane indicates that the synonyms were created successfully.
6. You can click *Go to Metadata page* where you can manage synonyms from the navigation pane.

Reference: Synonym Creation Parameters for Microsoft SQL Server

The following list describes the synonym creation parameters for which you can supply values.

Restrict Object Type to

Restrict candidates for synonym creation based on the selected object type(s): Tables, Views, External SQL Scripts, and any other supported objects.

Choosing *External SQL Scripts* from the drop-down list enables you to represent an SQL Query as a synonym for read-only reporting. A Synonym candidate can be any file that contains one (and only one) valid SQL Query and does not contain end-of-statement delimiters (";" or "/") and comments.

Depending on the adapter, you can further restrict your search by choosing

Important: If you select Stored Procedures as your object type, the input parameters will be a little different from those described here. For details, refer to [Creating a Report Against a Stored Procedure](#) on page 29.

Database selection

To specify a database from which you can select a table or other object, do one of the following:

- Check Use current database to use the database that has been set as the default database.
- Select a database from the Select database drop-down list, which lists all databases in the current DBMS instance.

Before selecting a database, if Use current database is checked, uncheck it.

To specify the intended database, choose from the Select database drop-down menu, which shows all databases on the targeted instance of Microsoft SQL Server. Selecting Default Database will retain the database set during connection configuration. If Default Database was not set during configuration, the database assigned to the active login on the SQL Server will be used as the default.

Filter by Owner/Schema and Object name

Selecting this option adds the Owner/Schema and Object Name parameters to the screen.

- Owner/Schema.** Type a string for filtering the selection, inserting the wildcard character (%) as needed at the beginning and/or end of the string. For example, enter: ABC% to select tables or views whose owner/schema begin with the letters ABC; %ABC to select tables or views whose owner/schema end with the letters ABC; %ABC% to select tables or views whose owner/schema contain the letters ABC at the beginning, middle, or end.
- Object name.** Type a string for filtering the object names, inserting the wildcard character (%) as needed at the beginning and/or end of the string. For example, enter: ABC% to select all objects whose names begin with the letters ABC; %ABC to select all whose names end with the letters ABC; %ABC% to select all whose names contain the letters ABC at the beginning, middle, or end.

Location of External SQL Scripts

If you specify *External SQL Scripts* in the *Restrict Object type to* field, these additional fields are displayed.

The following standard naming conventions apply for UNIX, IBM i IFS, and z/OS HFS:

- In the *Base Location* field, specify the physical directory location of the file that contains the SQL Query. You can type a directory name or click on the ellipsis. This opens the Select Base Location dialogue.
- In the *Document Name* field, enter the file name with or without wild card characters.
- In the *Document Extension* field, enter the extension of the script files to filter the list of candidates.

On IBM i, you can use alternative IFS naming conventions to access library members. The following entry illustrates this method:

- In the *Base Location* field, enter:

```
/QSYS.LIB/MYLIBRARY.LIB/MYSRC.FILE
```

- The *Document Extension* is understood to be MBR. You can enter this value explicitly or leave the input box blank.

During synonym generation, the adapter issues native API calls to obtain a list of elements in the select list and builds the Master File with a field for each element. The generated Access File references the location of the SQL script in the DATASET attribute, which contains the full path, including the name and extension of the file containing the SQL Query. For example,

```
DATASET=/ul/home2/apps/report3.sql
```

When a Web Query report is created, the SQL Query is used to access data.

Cardinality

Select the *Cardinality* check box to reflect the current cardinality (number of rows or tuples) in the table during metadata creation. Cardinality is used for equi-joins. The order of retrieval is based on the size (cardinality) of the table. Smaller tables are read first.

If the cardinality of the tables to be used in the application are dynamic, it may not be beneficial to choose this setting.

With foreign key

Select the *With foreign key* check box to include within this synonym every table related to the current table by a foreign key. The resulting multi-table synonym describes all of the foreign key relationships of this table.

Dynamic columns

To specify that the Master File created for the synonym should not contain column information, select the *Dynamic columns* check box.

If this option is selected, column data is retrieved dynamically from the data source at the time of the request.

Application

The default value is baseapp.

Prefix/Suffix

If you have tables with identical table names, assign a prefix or a suffix to distinguish them. For example, if you have identically named human resources and payroll tables, assign the prefix HR to distinguish the synonyms for the human resources tables. Note that the resulting synonym name cannot exceed 64 characters.

If all tables and views have unique names, leave the prefix and suffix fields blank.

Customize data type mappings

To change the data type mappings from their default settings, select this check box. The customizable mappings are displayed.

For information about customizable mappings, see [Microsoft SQL Server Data Type Support](#) on page 18.

Overwrite Existing Synonyms

To specify that this synonym should overwrite any earlier synonym with the same fully qualified name, select the *Overwrite existing synonyms* check box.

Note: The connected user must have operating system write privileges in order to recreate a synonym.

Default Synonym Name

This column displays the name that will be assigned to each synonym. To assign a different name, replace the displayed value.

Select tables

Select tables for which you wish to create synonyms:

- To select all tables in the list, click the *Select All* button.
- To select specific tables, select the corresponding check boxes.

Example: Sample Generated Synonym

DB2 Web Query Adapter for Microsoft SQL Server synonym comprises a Master File and an Access File. This is a synonym for the table nf29004.

Generated Master File nf29004.mas

```
FILE=DIVISION, SUFFIX=SQLMSS , $
SEGNAME=SEG1_4, SEGTYPE=S0 , $
FIELD=DIVISION4, DIVISION4, I9, I4, MISSING=OFF , $
FIELD=DIVISION_NA4, DIVISION_NA4, A25, A25, MISSING=ON , $
FIELD=DIVISION_HE4, DIVISION_HE4, I9, I4, MISSING=ON , $
```

Generated Access File nf29004.acx

```
SEGNAME=SEG1_4, TABLENAME=edaqa.nf29004,
CONNECTION=connmss, KEYS=1, WRITE=YES, $
```

Reference: Mapping Microsoft SQL Table Comments Into a Synonym

When you generate a synonym for a Microsoft SQL table, the adapter maps comments as follows:

- ❑ MS SQL Server table comments (if present) are mapped to the REMARKS attribute in the Master File synonym.
- ❑ MS SQL Server column comments (if present) are mapped to the DESCRIPTION attribute in the Master File synonym.

Both Unicode and non-Unicode comments are supported.

Reference: Access File Keywords

This chart describes the keywords in the Access File.

Keyword	Description
SEGNAME	Value must be identical to the SEGNAME value in the Master File.
TABLENAME	Identifies the Microsoft SQL Server table. The table name can be fully qualified as follows: <code>TABLENAME=[[database.]owner.]table</code>

Keyword	Description
CONNECTION	<p>Indicates a previously declared connection. The syntax is:</p> <p><i>CONNECTION=connection</i></p> <p>CONNECTION=' ' indicates access to the local database server.</p> <p>Absence of the CONNECTION attribute indicates access to the default database server.</p>
KEYS	<p>Indicates how many columns constitute the primary key for the table. Range is 0 to 64. Corresponds to the first <i>n</i> fields in the Master File segment.</p>
WRITE	<p>Specifies whether write operations are allowed against the table.</p>
KEYFLD IXFLD	<p>Supply the names of the primary key and foreign key fields that implement the relationships established by the multi-table Master File. Together, KEYFLD and IXFLD identify the field shared by a related table pair.</p> <ul style="list-style-type: none"> ❑ KEYFLD is the FIELDNAME of the common column from the parent table. ❑ IXFLD is the FIELDNAME of the common column from the related table. <p>KEYFLD and IXFLD must have the same data type. It is recommended, but not required, that their lengths also be the same.</p> <p>Note: An RDBMS index on both the KEYFLD and IXFLD columns provides the RDBMS with a greater opportunity to produce efficient joins. The columns must have the same data type. If their length is the same, the RDBMS handles the join more efficiently.</p>
AUTO INCREMENT	<p>Set to Yes to allow the auto increment feature.</p>
START	<p>Initial value in incrementing sequence.</p>
INCREMENT	<p>Increment interval.</p>
INDEX_NAME INDEX_UNIQUE INDEX_COLUMNS INDEX_ORDER	<p>Indicate a name of the index in a database, uniqueness, name, and order of the indexed column(s).</p>

Reference: Managing Synonyms

Once you have created a synonym, you can right-click the synonym name in the Adapter navigation pane to access the following options.

Option	Description
Open	Opens the Master File for viewing and editing using a graphical interface. If an Access file is used it will be also available.
Edit as Text	Enables you to view and manually edit the Master File synonym. Note: To update the synonym, it is strongly recommended that you use the graphical interface provided by the <i>Open</i> option, rather than manually editing the Master File.
Edit Access File as Text	Enables you to view and manually edit the Access File synonym. Note: This option is available only when an Access File is created as part of the synonym.
Sample Data	Retrieves up to 20 rows from the associated data source.
Data Profiling	Data Profiling provides the data characteristics for synonym columns. Alphanumeric columns provide the count of distinct values, total count, maximum, minimum, average length, and number of nulls. Numeric columns provide the count of distinct values, total count, maximum, minimum, average value, and number of nulls.
Refresh Synonym (if applicable)	Regenerates the synonym. Use this option if the underlying object has been altered.

Option	Description
Data Management	<p>Followed by these options, if applicable:</p> <p>Recreate DBMS Table. Recreates the data source table. You are asked to confirm this selection before the table is regenerated. (Note that the table will be dropped and recreated. During the process, data may be lost.)</p> <p>Delete All Data. Deletes all existing data. You are asked to confirm this selection before the data is deleted.</p> <p>Insert Sample Data. Inserts specified number of sample records, populating all fields with counter values.</p> <p>Reorganize. Recreates the data source table preserving original data.</p> <p>Note: This option is not available in the Web Console.</p>
Impact Analysis	<p>Generates reports on procedures, synonyms, and columns that provide information on the flows/stored procedures available on a particular server, and the synonyms and columns they use. These reports enable you to evaluate changes before they are made by showing which components will be affected.</p>
Copy	<p>Copies the synonym to clipboard.</p>
Delete	<p>Deletes the synonym. You are asked to confirm this selection before the synonym is deleted.</p>
Cut	<p>Deletes the synonym and places it in clipboard.</p>
Properties	<p>Displays the properties of the synonym, including physical location, last modified date, description, and privileges.</p>

Microsoft SQL Server Data Type Support

Reference:

Data Type Support for Unicode

The following tables describe how the server maps Microsoft SQL Server data types.

Microsoft SQL Server Data Type	Server Data Type		Remarks
	USAGE	ACTUAL	
Date-Time Data Types			
TIMESTAMP	A16	A16	Supported as Read-only.
DATE	YYMD	DATE	Range: 0001-01-01 through 9999-12-31. Available only for MS SQL 2008.
TIME	HHIS	HHIS	Range: 00:00:00.0000000 through 23:59:59.9999999. Available only for MS SQL 2008.
DATETIMEOFFSET	HYYMDS	HYYMDS	Range: date portion 0001-01-01 through 9999-12-31, time portion 00:00:00.0000000 through 23:59:59.9999999, time zone offset portion -14:00 through +14:00. Available only for MS SQL 2008.
DATETIME	HYYMDs	HYYMDs	Range: 1/1/1753 to 12/31/9999.
DATETIME2	HYYMDm	HYYMDm	Range: date portion 0001-01-01 through 9999-12-31, time portion 00:00:00.0000000 through 23:59:59.9999999. Available only for MS SQL 2008.
SMALLDATETIME	HYYMDI	HYYMDI	Range: 1/1/1900 thru 6/6/2079.
Numeric Data Types			
INT	I11	I4	Range: -2^{31} to $2^{31} - 1$.
BIGINT	P20	P10	Range: -2^{63} to $2^{63} - 1$.
SMALLINT	I6	I4	Range: -2^{15} to $2^{15} - 1$.
TINYINT	I6	I4	Range: 0 to 255.
BIT	I11	I4	-1 for True and 0 for False.

Microsoft SQL Server Data Type	Server Data Type		Remarks
	USAGE	ACTUAL	
DECIMAL (p,s)	Pn.m	Pk	<p>p is an integer between 1 and 38. s is an integer between 0 and p.</p> <p>If s is 0 and p is between 1 and 31, $n = p + 1$.</p> <p>If s is 0 and p is between 32 and 38, $n = 32$.</p> <p>If s is greater than 0 and p is between 1 and 31, $n = p + 2$ and $m = s$.</p> <p>If s is greater than 0 and p is between 32 and 38, $n = 33$ and $m = 31$.</p> <p>If p is between 1 and 31, $k = (p / 2) + 1$.</p> <p>If p is between 32 and 38, $k = 16$.</p> <p>Note: If the column is nullable, p is greater than or equal to 8.</p>
NUMERIC (p,s)	Pn.m	Pk	<p>p is an integer between 1 and 38. s is an integer between 0 and p.</p> <p>If s is 0 and p is between 1 and 31, $n = p + 1$.</p> <p>If s is 0 and p is between 32 and 38, $n = 32$.</p> <p>If s is greater than 0 and p is between 1 and 31, $n = p + 2$ and $m = s$.</p> <p>If s is greater than 0 and p is between 32 and 38, $n = 33$ and $m = 31$.</p> <p>If p is between 1 and 31, $k = (p / 2) + 1$.</p> <p>If p is between 32 and 38, $k = 16$.</p> <p>Note: If the column is nullable, p is greater than or equal to 8.</p>
MONEY	P21.4	P10	Range: -2^{63} to $2^{63} - 1$.

Microsoft SQL Server Data Type	Server Data Type		Remarks
	USAGE	ACTUAL	
SMALLMONEY	P12.4	P8	Range: -214,748.3648 to 214,748.3647.
FLOAT	D20.2	D8	Range: -1.79E+308 to 1.79E+308.
REAL	D20.2	D8	Range: -3.40E+38 to 3.40E+38.
LOB Data Types			
TEXT	TX50	TX	Character LOB data. Maximum storage size is $2^{31} - 1$ bytes.
NTEXT	TX50	TX	Character LOB data. Maximum storage size is $2^{31} - 1$ bytes.
VARCHAR (MAX)	TX50	TXV	Character LOB data type is a replacement for TEXT. Maximum storage size is $2^{31} - 1$ bytes.
NVARCHAR (MAX)	TX50	TX	Character LOB data type is a replacement for NTEXT. Maximum storage size is $2^{31} - 1$ bytes.
IMAGE	BLOB	BLOB	Length: $2^{31} - 1$. Supported through the Reporting Server API.
VARBINARY(MAX)	BLOB	BLOB	Replacement for IMAGE. Length: $2^{31} - 1$. Supported through the Reporting Server API.
Other Data Types			
UNIQUEIDENTIFIER (GUID)	A38	A38	

The following table lists how the non-Unicode server maps Character data types. This mapping can be changed based on the value of LONGCHAR. The default value is ALPHA.

Microsoft SQL Server Data Type	Remarks	LONGCHAR ALPHA		LONGCHAR TEXT	
		USAGE	ACTUAL	USAGE	ACTUAL
CHAR (<i>n</i>)	<i>n</i> is an integer between 1 and 256.	<i>An</i>	<i>An</i>	<i>An</i>	<i>An</i>
	<i>n</i> is an integer between 257 and 8000.	<i>An</i>	<i>An</i>	TX50	TX
NCHAR (<i>n</i>)	<i>n</i> is an integer between 1 and 128.	<i>An</i>	<i>An</i>	<i>An</i>	<i>An</i>
	<i>n</i> is an integer between 129 and 4000.	<i>An</i>	<i>An</i>	TX50	TX
VARCHAR (<i>n</i>)	<i>n</i> is an integer between 1 and 256.	<i>AnV</i>	<i>AnV</i>	<i>AnV</i>	<i>AnV</i>
		<i>AnV</i>	<i>AnV</i>	TX50	TX
NVARCHAR (<i>n</i>)	<i>n</i> is an integer between 1 and 128.	<i>AnV</i>	<i>AnV</i>	<i>AnV</i>	<i>AnV</i>
	<i>n</i> is an integer between 129 and 4000.	<i>AnV</i>	<i>AnV</i>	TX50	TX
BINARY (<i>n</i>)	<i>n</i> is an integer between 1 and 8000, $m = 2 * n$.	<i>Am</i>	<i>Am</i>	TX50	TX
VARBINARY (<i>n</i>)	<i>n</i> is an integer between 1 and 8000, $m = 2 * n$.	<i>AmV</i>	<i>AmV</i>	TX50	TX
SQL_VARIANT		A4000V	A4000V	TX50	TX

Note: The main purpose of the LONGCHAR setting is to provide compatibility with previous releases of the server. This parameter was designed to control processing of DBMS Character data types and was never intended for DBMS LOB.

Reference: Data Type Support for Unicode

The following table describes how the Unicode server maps Character data types. The server operates in character semantic when configured for Unicode. The LONGCHAR does not affect mapping in this case.

Microsoft SQL Server Data Type	Data Type		Remarks
	USAGE	ACTUAL	
CHAR (<i>n</i>) double-byte code page	AnV	AnV	<i>n</i> is an integer between 1 and 8000. This data type does not support Unicode. However, character columns such as CHAR that had been created with different code pages can be read at the same time only if the server has been configured for Unicode.
NCHAR (<i>n</i>)	An	An	<i>n</i> specifies the number of characters, and is an integer between 1 and 4000. This data type supports Unicode.
VARCHAR (<i>n</i>)	AnV	AnV	<i>n</i> is an integer between 1 and 8000. This data type does not support Unicode. However, character columns such as VARCHAR that had been created with different code pages can be read at the same time only if the server has been configured for Unicode.
NVARCHAR (<i>n</i>)	AnV	AnV	<i>n</i> specifies the number of characters, and is an integer between 1 and 4000. This data type supports Unicode.

Enabling National Language Support

How to:

Enable National Language Support

The parameter NCHAR indicates whether the character set is single-byte, double-byte, or triple-byte. The NCHAR setting affects the mapping of NCHAR and NVARCHAR data types.

The following chart lists data type mappings based on the value of NCHAR.

Microsoft SQL Server Data Type	Remarks	NCHAR SBCS		NCHAR DBCS		NCHAR TBCS	
		USAGE	ACTUAL	USAGE	ACTUAL	USAGE	ACTUAL
NCHAR (n)	n is an integer between 1 and 4000 d = 2 * n t = 3 * n	An	An	Ad	Ad	At	At
NVARCHAR (n)	n is an integer between 1 and 4000 d = 2 * n t = 3 * n	An	An	Ad	Ad	At	At

Syntax: How to Enable National Language Support

You change this setting by expanding the Domains and Reports folders, then right-clicking a subfolder and choosing *Metadata*. Right-click MS SQL Server 200x and select *Change Settings*. The Change Settings pane opens.

SQLMSS

Indicates the adapter. You can omit this value if you previously issued the `SQLENGINE` command.

SBCS

Indicates a single-byte character set. SBCS is the default value.

DBCS

Indicates a double-byte character set.

TBCS

Indicates a triple-byte character set.

Support of Read-Only Fields

`CREATE SYNONYM` creates a field description with `FIELDTYPE=R` for Microsoft SQL Server columns created as `TIMESTAMP` or columns with the `IDENTITY` attribute. These fields are read-only.

Reporting Against a Microsoft SQL Server Stored Procedure

In this section:

- Generating a Synonym for a Stored Procedure
- Creating a Report Against a Stored Procedure

You can use a reporting tool, such as a SELECT statement or TABLE command, to execute Microsoft SQL Server stored procedures and report against the procedure output parameters and answer set. Among the benefits of this method of executing a stored procedure are:

- ❑ The retrieval of output parameters: OUT parameters, and INOUT parameters in OUT mode, as well as the answer set. (Other methods of invocation retrieve the answer set only.)
- ❑ The ease with which you can process, format, and display output parameters and the answer set, using TABLE and other reporting tools.

To report against a stored procedure:

- 1. Generate a synonym** for the stored procedure answer set, as described in [Generating a Synonym for a Stored Procedure](#) on page 25.
- 2. Create a report procedure**, as described in [Creating a Report Against a Stored Procedure](#) on page 29.
- 3. Run the report.** This executes the stored procedure and reports against any output parameters (OUT and INOUT in OUT mode), and any answer set fields, specified in the report.

Generating a Synonym for a Stored Procedure

Reference:

- Synonym Creation Parameters for Stored Procedures

A synonym describes the stored procedure parameters and answer set.

An answer set structure may vary depending on the input parameter values that are provided when the procedure is executed. Therefore, you need to generate a separate synonym for each set of input parameter values that will be provided when the procedure is executed at run time. For example, if users may execute the stored procedure using three different sets of input parameter values, you need to generate three synonyms, one for each set of values. (Unless noted otherwise, *input parameters* refers to IN parameters and to INOUT parameters in IN mode.)

There is an exception. If you know the internal logic of the procedure, and are certain which range of input parameter values will generate each answer set structure returned by the procedure, you can create one synonym for each answer set structure, and for each synonym simply provide a representative set of the input parameter values necessary to return that answer set structure.

A synonym includes the following segments:

- ❑ INPUT, which describes any IN parameters and INOUT parameters in IN mode.
If there are no IN parameters or INOUT parameters in IN mode, the segment describes a single dummy field.
- ❑ OUTPUT, which describes any OUT parameters and INOUT parameters in OUT mode.
If there are no OUT parameters or INOUT parameters in OUT mode, the segment is omitted.
- ❑ ANSWERSET n , one for each answer set.
If there is no answer set, the segment is omitted.

Example: Synonym for Microsoft SQL Server Stored Procedure CustOrders

The following synonym describes a Microsoft SQL Server stored procedure with one input parameter, one output parameter, and one answer set containing four variables.

The Master File synonym is:

```
FILENAME=CUSTORDERS, SUFFIX=SQLMSS , $
  SEGMENT=INPUT, SEGTYPE=S0, $
    FIELDNAME=@CUSTOMERID, ALIAS=P0001, USAGE=A5, ACTUAL=A5,
    MISSING=ON, ACCESS_PROPERTY=(NEED_VALUE), $
  SEGMENT=OUTPUT, SEGTYPE=S0, PARENT=INPUT, $
    FIELDNAME=@RETURN_VALUE, ALIAS=P0000, USAGE=I11, ACTUAL=I4, $
  SEGMENT=ANSWERSET1, SEGTYPE=S0, PARENT=INPUT, $
    FIELDNAME=ORDERID, ALIAS=OrderID, USAGE=I11, ACTUAL=I4, $
    FIELDNAME=ORDERDATE, ALIAS=OrderDate, USAGE=HYYMDs, ACTUAL=HYYMDs,
    MISSING=ON, $
    FIELDNAME=REQUIREDDATE, ALIAS=RequiredDate, USAGE=HYYMDs,
    ACTUAL=HYYMDs, MISSING=ON, $
    FIELDNAME=SHIPPEDDATE, ALIAS=ShippedDate, USAGE=HYYMDs,
    ACTUAL=HYYMDs, MISSING=ON, $
```

The Access File synonym is:

```
SEGNAME=INPUT, CONNECTION=ITarget, STPNAME=Northwind.dbo.CustOrders, $
SEGNAME=OUTPUT, STPRESORDER=0, $
SEGNAME=ANSWERSET1, STPRESORDER=1, $
```

Reference: Synonym Creation Parameters for Stored Procedures

The following list describes the synonym creation parameters for which you can supply values.

Restrict Object Type to

Select *Stored Procedures*.

Database selection

To specify a database from which you can select a table or other object, do one of the following:

- Check Use current database to use the database that has been set as the default database.
- Select a database from the Select database drop-down list, which lists all databases in the current DBMS instance.

Before selecting a database, if Use current database is checked, uncheck it.

Filter by Owner/Schema and Object name

Selecting this option adds the Owner/Schema and Object Name parameters to the screen.

- Owner/Schema.** Type a string for filtering the selection, inserting the wildcard character (%) as needed at the beginning and/or end of the string. For example, enter: ABC% to select tables or views whose owner/schema begin with the letters ABC; %ABC to select tables or views whose owner/schema end with the letters ABC; %ABC% to select tables or views whose owner/schema contain the letters ABC at the beginning, middle, or end.
- Object name.** Type a string for filtering the object names, inserting the wildcard character (%) as needed at the beginning and/or end of the string. For example, enter: ABC% to select all objects whose names begin with the letters ABC; %ABC to select all whose names end with the letters ABC; %ABC% to select all whose names contain the letters ABC at the beginning, middle, or end.

Note: For DB2, this applies to all platforms except IBM i.

Select

Select a procedure. You may only select one procedure at a time since each procedure will require unique input in the Values box on the next synonym creation pane.

Name

The name of the synonym, which defaults to the stored procedure name.

Application

The default value is baseapp.

Prefix/Suffix

If you have stored procedures with identical names, assign a prefix or a suffix to distinguish their corresponding synonyms. Note that the resulting synonym name cannot exceed 64 characters.

If all procedures have unique names, leave the prefix and suffix fields blank.

Overwrite Existing Synonyms

To specify that this synonym should overwrite any earlier synonym with the same fully qualified name, select the *Overwrite existing synonyms* check box.

Note: The connected user must have operating system write privileges in order to recreate a synonym.

Customize data type mappings

To change the data type mappings from their default settings, select this check box. The customizable mappings are displayed.

For information about customizable mappings, see [Microsoft SQL Server Data Type Support](#) on page 18.

Values

Select the check box for every parameter displayed for the specified procedure.

Note the following before you enter parameter values: if the procedure you selected has input parameters (IN parameters and/or INOUT parameters in IN mode), you will be prompted to enter values for them. However, the need for an explicit Value entry depends on the logic of the procedure and the data structures it produces. Therefore, while you must check the parameter box, you may not need to enter a value. Follow these guidelines:

- Explicit input values (and separate synonyms) are required when input parameter values cause answer sets with different data structures, which vary depending on the input parameters provided.
- Explicit input values are not required when you know the procedure's internal logic and are certain that it always produces the same data structure. In this situation, only one synonym needs to be created and you can leave the Value input blank for synonym creation purposes.

If a Value is required, enter it without quotes. Any date, date-time, and timestamp parameters must have values entered in an ISO format. Specify the same input parameters that will be provided when the procedure is executed at run time if it is a procedure that requires explicit values.

Creating a Report Against a Stored Procedure

How to:

Report Against a Stored Procedure Using the TABLE Command

Report Against a Stored Procedure Using SELECT

You can report against a stored procedure answer set using the same facilities you use to report against a database table:

- ❑ **SQL SELECT statement.** For syntax, see [How to Report Against a Stored Procedure Using SELECT](#) on page 30.
- ❑ **TABLE and GRAPH commands.** For syntax, see [How to Report Against a Stored Procedure Using the TABLE Command](#) on page 29.

When joining from or to a stored procedure answer set, you can:

- ❑ **Join from** only OUTPUT and ANSWERSET segments in a host file.
- ❑ **Join to** only INPUT segments in a cross-referenced file.

Syntax: How to Report Against a Stored Procedure Using the TABLE Command

To execute a stored procedure using the TABLE command, use the following syntax

```
TABLE FILE synonym
PRINT [parameter [parameter] ... | *]
[IF in-parameter EQ value]
.
.
.
END
```

where:

synonym

Is the synonym of the stored procedure you want to execute.

parameter

Is the name of a parameter whose values you want to display in the report. You can specify input parameters, output parameters, or input and output parameters.

If the stored procedure does not require parameters, specify an asterisk (*). This displays a dummy segment, created when the synonym is generated, to satisfy the structure of the SELECT statement.

*

Indicates that you want to display all indicated parameters, or that there are no required parameters.

IF

Is an IF or WHERE keyword. Use this to pass a value to an IN parameter or an INOUT parameter in IN mode.

in-parameter

Is the name of an IN parameter, or INOUT parameter in IN mode, to which you want to pass a value.

Note: The length of in-parameters cannot exceed 1000 characters if the adapter is configured for Unicode support.

value

Is the value you are passing to a parameter.

Syntax: How to Report Against a Stored Procedure Using SELECT

```
SQL
SELECT [parameter [,parameter] ... | *] FROM synonym
[WHERE in-parameter = value]
.
.
.
END
```

where:

synonym

Is the synonym of the stored procedure that you want to execute.

parameter

Is the name of a parameter whose values you want to display in the report. You can specify input parameters, output parameters, or input and output parameters.

If the stored procedure does not require parameters, enter an asterisk (*) in the syntax. This displays a dummy segment, created during synonym generation, to satisfy the structure of the SELECT statement.

*

Indicates that you want to display all indicated parameters, or that there are no required parameters.

WHERE

Is used to pass a value to an IN parameter or an INOUT parameter in IN mode.

You must specify the value of each parameter on a separate line.

in-parameter

Is the name of an IN parameter, or INOUT parameter in IN mode, to which you want to pass a value.

value

Is the value you are passing to a parameter.

Customizing the Microsoft SQL Server Environment

In this section:

Specifying the Cursor Type

Activating NONBLOCK Mode

Obtaining the Number of Rows Updated or Deleted

Controlling Transactions

Specifying the Transaction Isolation Level

The DB2 Web Query Adapter for Microsoft SQL Server provides several parameters for customizing the environment and optimizing performance. This topic provides an overview of customization options.

Specifying the Cursor Type

How to:

Specify the Cursor Type

You can use the CURSORS command to specify the type of cursors for retrieval.

Syntax: How to Specify the Cursor Type

You change this setting by expanding the Domains and Reports folders, then right-clicking a subfolder and choosing *Metadata*. Right-click MS SQL Server 200x and select *Change Settings*. The Change Settings pane opens.

CLIENT

Uses Microsoft SQL Server client-side cursors for retrieving data. Client-side cursors normally demonstrate the best performance for data retrieval and benefit the Microsoft SQL Server process. However, except in TRANSACTIONS AUTOCOMMITTED mode, using client-side cursors prevents a server agent from simultaneously reading more than one answer set from the same instance of Microsoft SQL Server.

SERVER

Uses Microsoft SQL Server server-side cursors for retrieving data. Server-side cursors demonstrate lower performance than client cursors. However, setting a high FETCHSIZE factor (100 is the adapter default) improves performance dramatically making them almost as fast as client-side cursors. Client-side cursors are recommended wherever possible to take the load off the Microsoft SQL Server process.

blank

Uses client-side cursors in TRANSACTIONS AUTOCOMMITTED mode and server-side cursors otherwise. This value is the default.

Activating NONBLOCK Mode

How to:

Activate NONBLOCK Mode

The Adapter for Microsoft SQL Server has the ability to issue calls in NONBLOCK mode. The default behavior is BLOCK mode.

This feature allows the adapter to react to a client request to cancel a query while the adapter is waiting on engine processing. This wait state usually occurs during SQL parsing, before the first row of an answer set is ready for delivery to the adapter or while waiting for access to an object that has been locked by another application.

Syntax: **How to Activate NONBLOCK Mode**

You change this setting by expanding the Domains and Reports folders, then right-clicking a subfolder and choosing *Metadata*. Right-click MS SQL Server 200x and select *Change Settings*. The Change Settings pane opens.

SQLMSS

Indicates the adapter. You can omit this value if you previously issued the `SQLENGINE` command.

n

Is a positive numeric number. 0 is the default value, which means that the adapter will operate in BLOCK mode. A value of 1 or greater activates the NONBLOCK calling and specifies the time, in seconds, that the adapter will wait between each time it checks to see if the:

- Query has been executed.
- Client application has requested the cancellation of a query.
- Kill Session button on the Web Console is pressed.

Note: A value of 1 or 2 should be sufficient for normal operations.

Obtaining the Number of Rows Updated or Deleted**How to:**

Obtain the Number of Rows Updated or Deleted

PASSRECS returns the number of rows affected by a successfully executed SQL Passthru INSERT, UPDATE, or DELETE command.

Syntax: **How to Obtain the Number of Rows Updated or Deleted**

You change this setting by expanding the Domains and Reports folders, then right-clicking a subfolder and choosing *Metadata*. Right-click MS SQL Server 200x and select *Change Settings*. The Change Settings pane opens.

SQLMSS

Indicates the adapter. You can omit this value if you previously issued the `SQLENGINE` command.

[ON](#)

Provides the number of rows affected in the application program SCB count member after the successful execution of an SQL Passthru INSERT, UPDATE, or DELETE command. ON is the default value.

[OFF](#)

Provides no information after the successful execution of an SQL Passthru INSERT, UPDATE, or DELETE command.

Controlling Transactions

How to:

Control Transactions

The TRANSACTIONS command to controls how the adapter handles transactions.

Syntax: **How to Control Transactions**

You change this setting by expanding the Domains and Reports folders, then right-clicking a subfolder and choosing *Metadata*. Right-click MS SQL Server 200x and select *Change Settings*. The Change Settings pane opens.

[SQLMSS](#)

Indicates the adapter. You can omit this value if you previously issued the SQLENGINE command.

[LOCAL](#)

Indicates that the adapter implicitly starts a local transaction on each of the connections where any work is performed. At the time of COMMIT or ROLLBACK, or at the end of the server session, the adapter commits or aborts the work on each connection consecutively. LOCAL is the default value.

[DISTRIBUTED](#)

Indicates that the adapter implicitly invokes Microsoft Distributed Transactions Coordinator (DTC) to create a single distributed transaction within which to perform all work on all the connections. At the time of COMMIT or ROLLBACK, or at the end of the server session, the adapter invokes DTC to execute the two-phase commit or rollback protocol. For this purpose, the DTC service must be started on the machine where the server is running and also on all the machines where involved instances of Microsoft SQL Server reside.

This mode is recommended for read-write applications that perform updates on multiple connections simultaneously.

AUTOCOMMITTED

Indicates that each individual operation with Microsoft SQL Server is immediately committed (if successful) or rolled back (in case of errors) by the SQL Server. This is recommended for read-only applications for performance considerations. It is not recommended for read-write applications because in this mode it is impossible to roll back a logical unit of work that consists of several operations.

Specifying the Transaction Isolation Level

How to:

Specify Transaction Isolation Level

You can specify the transaction isolation level from the Web Console.

Syntax: **How to Specify Transaction Isolation Level**

You change this setting by expanding the Domains and Reports folders, then right-clicking a subfolder and choosing *Metadata*. Right-click MS SQL Server 200x and select *Change Settings*. The Change Settings pane opens.

The available parameters are:

RU

Sets the transaction isolation level to Read Uncommitted.

RC

Sets the transaction isolation level to Read Committed.

RR

Sets the transaction isolation level to Repeatable Read.

SE

Sets the transaction isolation level to Serializable Read.

CH

Sets the transaction isolation level to Chaos.

CS

Sets the transaction isolation level to Cursor Stability, which is a synonym for Read Committed.

Optimization Settings

In this section:

- Optimizing Requests
- Optimizing Requests Containing IF-THEN-ELSE Virtual Fields
- Optimizing Requests if a Virtual Field Contains Null Values

Adapter optimization allows the RDBMS to perform the work for which it is best suited, reducing the volume of RDBMS-to-server communication and improving response time. It also enables the RDBMS to exploit its own internal optimization techniques.

Optimizing Requests

How to:

- Optimize Requests

Reference:

- SQL Generation in Optimization Examples

The adapter can optimize DML requests by creating SQL statements that take advantage of RDBMS join, sort, and aggregation capabilities.

You change this setting by expanding the Domains and Reports folders, then right-clicking a subfolder and choosing *Metadata*. Right-click SQLMSS and select *Change Settings*. The Change Settings pane opens.

Syntax: **How to Optimize Requests**

SQLMSS

Is the target RDBMS. You can omit this value if you previously issued the SQLENGINE command.

SQLJOIN

Is a synonym for OPTIMIZATION.

setting

Is the optimization setting. Valid values are as follows:

ON instructs the adapter to create SQL statements that take advantage of RDBMS join, sort, and aggregation capabilities. Note that the multiplicative effect may disable optimization in some cases. However, misjoined unique segments and multiplied lines in PRINT-based and LIST-based report requests do not disable optimization. This is the default.

OFF instructs the adapter to create SQL statements for simple data retrieval from each table. The server handles all aggregation, sorting, and joining in your address space or virtual machine to produce the report.

Both OPTIMIZATION settings produce the same report.

Example: SQL Requests Passed to the RDBMS With Optimization OFF

This example demonstrates SQL statements generated without optimization. The report request joins tables EMPINFO and FUNDTRAN with trace components SQLAGGR and STMTRACE allocated.

When optimization is disabled, the data adapter generates two SELECT statements. The first SELECT retrieves any rows from the EMPINFO table that have the value MIS in the DEPARTMENT column. For each EMPINFO row, the second SELECT retrieves rows from the cross-referenced FUNDTRAN table, resolving the parameter marker (?, :000n, or :H, depending on the RDBMS) with the value of the host field (EMP_ID). Both SELECT statements retrieve answer sets, but the server performs the join, sort, and aggregation operations:

```
SQL QLMSS SET OPTIMIZATION OFF
JOIN EMP_ID IN EMPINFO TO ALL WHO IN FUNDTRAN AS J1
TABLE FILE EMPINFO
  SUM AVE.CURRENT_SALARY ED_HRS BY WHO BY LAST_NAME
  IF DEPARTMENT EQ 'MIS'
END
```

In a trace operation, you will see the following output:

```
(FOC2510) FOCUS-MANAGED JOIN SELECTED FOR FOLLOWING REASON(S):
(FOC2511) DISABLED BY USER
(FOC2590) AGGREGATION NOT DONE FOR THE FOLLOWING REASON:
(FOC2592) RDBMS-MANAGED JOIN HAS BEEN DISABLED
  SELECT T1.EID,T1.LN,T1.DPT,T1.CSAL,T1.OJT
    FROM 'USER1'.'EMPINFO' T1 WHERE (T1.DPT = 'MIS') FOR FETCH ONLY;
  SELECT T2.EID FROM 'USER1'.'FUNDTRAN' T2 WHERE (T2.EID = ?)
    FOR FETCH ONLY;
```

Example: SQL Requests Passed to the RDBMS With Optimization ON

With optimization enabled, the data adapter generates one SELECT statement that incorporates the join, sort, and aggregation operations. The RDBMS manages and processes the request. The server only formats the report.

```
SQL SQLMSS SET OPTIMIZATION ON
JOIN EMP_ID IN EMPINFO TO ALL WHO IN FUNDTRAN AS J1
TABLE FILE EMPINFO
SUM AVE.CURRENT_SALARY ED_HRS BY WHO BY LAST_NAME
IF DEPARTMENT EQ 'MIS'
END
```

In a trace operation, you will see the following output:

```
AGGREGATION DONE ...
SELECT T2.EID,T1.LN, AVG(T1.CSAL), SUM(T1.OJT)
FROM 'USER1'.'EMPINFO' T1,'USER1'.'FUNDTRAN' T2
WHERE (T2.EID = T1.EID) AND (T1.DPT = 'MIS')
GROUP BY T2.EID,T1.LN
ORDER BY T2.EID,T1.LN;
```

Reference: SQL Generation in Optimization Examples

There are minor differences in the specific SQL syntax generated for each RDBMS. However, the adapter messages are the same and the generated SQL statements are similar enough that most examples will illustrate the SQL syntax generated by any relational adapter.

Optimizing Requests Containing IF-THEN-ELSE Virtual Fields

How to:
Optimize Requests Containing IF-THEN-ELSE Virtual Fields

Reference:
SQL Limitations on Optimization of DEFINE Expressions

The adapter can optimize DML requests to the server that include virtual (DEFINE) fields created using IF-THEN-ELSE syntax. In certain cases, such DEFINE fields can be passed to the RDBMS as expressions, enhancing performance and minimizing the size of the answer set returned to the server.

You change this setting by expanding the Domains and Reports folders, then right-clicking a subfolder and choosing *Metadata*. Right-click SQLMSS and select *Change Settings*. The Change Settings pane opens.

Syntax: How to Optimize Requests Containing IF-THEN-ELSE Virtual Fields**SQLMSS**

Indicates the target RDBMS. You can omit this value if you previously issued the `SQLENGINE` command.

ON

Enables IF-THEN-ELSE optimization. This is the default.

OFF

Disables IF-THEN-ELSE optimization.

There is no guarantee that the SQL that is generated will improve performance for all requests. If you find that this feature does not improve performance, set `OPTIFTHENELSE OFF` to disable the feature.

IF-THEN-ELSE optimization applies to `SELECT` statements created as a result of requests and is subject to SQL limitations on optimization of `DEFINE` expressions.

Example: Using IF-THEN_ELSE Optimization Without Aggregation

Consider the following request:

```
SQL SQLMSS SET OPTIFTHENELSE ON
DEFINE FILE EMPINFO
DEF1 = IF (LAST_NAME EQ ' ') AND (FIRST_NAME EQ ' ')
        AND (DEPARTMENT EQ 'MIS') THEN 1 ELSE 0;
END
TABLE FILE EMPINFO
PRINT DEPARTMENT LAST_NAME FIRST_NAME
WHERE DEF1 EQ 1
END
```

The adapter generates an SQL request that incorporates the IF-THEN-ELSE condition corresponding to the `WHERE DEF1 EQ 1` test:

```
SELECT T1.'LN',T1.'FN',T1.'DPT' FROM USER1.'EMPINFO' T1
WHERE (((((T1.'LN' = ' ') AND (T1.'FN' = ' '))
        AND (T1.'DPT' = 'MIS'))));
```

Example: Using IF-THEN_ELSE Optimization With Aggregation

Consider the following request:

```
SQL SQLMSS SET OPTIFTHENELSE ON
DEFINE FILE EMPINFO
DEF2 = IF LAST_NAME EQ 'SMITH' THEN 1 ELSE IF LAST_NAME EQ 'JONES' THEN 2
      ELSE IF LAST_NAME EQ 'CARTER' THEN 3 ELSE 0;
END

TABLE FILE EMPINFO
SUM MAX.LAST_NAME IF DEF2 EQ 1
END
```

The adapter generates an SQL request that incorporates the IF-THEN-ELSE condition corresponding to the WHERE DEF2 EQ 1 test:

```
SELECT MAX(T1.'LN') FROM USER1.'EMPINFO' T1
WHERE (((T1.'LN' = 'SMITH')));

TABLE FILE EMPINFO
SUM MAX.LAST_NAME IF DEF2 EQ 2
END
```

The adapter generates an SQL request that incorporates the IF-THEN-ELSE condition corresponding to the WHERE DEF2 EQ 2 test:

```
SELECT MAX(T1.'LN') FROM USER1.'EMPINFO' T1
WHERE (((NOT (T1.'LN' = 'SMITH')) AND (T1.'LN' = 'JONES')));
```

Example: Using IF-THEN_ELSE Optimization With a Condition That Is Always False

```
SQL SQLMSS SET OPTIFTHENELSE ON
DEFINE FILE EMPINFO
DEF3 = IF FIRST_NAME EQ 'RITA' THEN 1 ELSE 0;
END

TABLE FILE EMPINFO
PRINT FIRST_NAME
IF DEF3 EQ 2
END
```

Because DEF3 EQ 2 will never be true, the adapter passes the WHERE test 1=0 (which is always false) to the RDBMS, returning zero records from the RDBMS:

```
SELECT T1.'FN' FROM USER1.'EMPINFO' T1 WHERE (1 = 0);
```


Reference: SQL Limitations on Optimization of DEFINE Expressions

Since the Web Query reporting language is more extensive than native SQL, the data adapter cannot pass certain DEFINE expressions to the RDBMS for processing. The data adapter does not offload DEFINE-based aggregation and record selection if the DEFINE includes:

- ❑ User-written subroutines.
- ❑ Self-referential expressions, such as:
`X=X+1;`
- ❑ EDIT functions for numeric-to-alpha or alpha-to-numeric field conversions.
- ❑ DECODE functions for field value conversions.
- ❑ Relational operators INCLUDES and EXCLUDES.
- ❑ Web Query subroutines ABS, INT, MAX, MIN, LOG, and SQRT.

Note: Do not confuse the Web Query user-written subroutines MAX and MIN with the MAX. and MIN. prefix operators. DEFINE fields cannot include prefix operators.

- ❑ Expressions involving fields with ACTUAL=DATE, except for the subtraction of one DATE field from another and all logical expressions on DATE fields.
- ❑ Date-time manipulation handled by the FOCUS date-time functions is not converted to SQL.

In addition, IF-THEN-ELSE optimization does not support the following features:

- ❑ Any type of DECODE expression.
- ❑ STATIC SQL.
- ❑ IF/WHERE DDNAME.
- ❑ Partial date selection.

Optimizing Requests if a Virtual Field Contains Null Values

Reference:

SQL Limitations on Optimization of DEFINE Expressions

The OPTNOAGGR command provides finely-tuned control of adapter behavior for optimization. Users who for any reason wish to prevent passing aggregation to the RDBMS can use this command. An example of such a reason might be where NULL values occur in aggregated data with calculations. The OPTNOAGGR command causes the adapter to generate SQL without passing aggregation to the DBMS. Aggregation is instead performed internally by the server while JOIN and SORT operations are handled by the RDBMS.

If any DEFINE field contains calculations with NULL fields then such operations cannot be translated to SQL and pass to DBMS because always return NULL. It has to be processed by Web Query.

This can be achieved by setting OPTIMIZATION to OFF.

However, in some cases it is preferable to use the off-load JOIN and SORT operation to DBMS for better performance while leaving AGGREGATION to Web Query.

Reference: [SQL Limitations on Optimization of DEFINE Expressions](#)

Since the Web Query reporting language is more extensive than native SQL, the data adapter cannot pass certain DEFINE expressions to the RDBMS for processing. The data adapter does not offload DEFINE-based aggregation and record selection if the DEFINE includes:

- ❑ User-written subroutines.
- ❑ Self-referential expressions, such as:

```
X=X+1;
```
- ❑ EDIT functions for numeric-to-alpha or alpha-to-numeric field conversions.
- ❑ DECODE functions for field value conversions.
- ❑ Relational operators INCLUDES and EXCLUDES.
- ❑ Web Query subroutines ABS, INT, MAX, MIN, LOG, and SQRT.

Note: Do not confuse the Web Query user-written subroutines MAX and MIN with the MAX. and MIN. prefix operators. DEFINE fields cannot include prefix operators.

- ❑ Expressions involving fields with ACTUAL=DATE, except for the subtraction of one DATE field from another and all logical expressions on DATE fields.

- ❑ Date-time manipulation handled by the Web Query date-time functions is not converted to SQL.

In addition, IF-THEN-ELSE optimization does not support the following features:

- ❑ Any type of DECODE expression.
- ❑ STATIC SQL.
- ❑ IF/WHERE DDNAME.
- ❑ Partial date selection.

Calling a Microsoft SQL Server Stored Procedure Using SQL Passthru

How to:

Invoke a Stored Procedure

Reference:

Capturing Application Errors in Stored Procedures

Microsoft SQL Server stored procedures are supported using SQL Passthru. These procedures need to be developed within Microsoft SQL Server using the CREATE PROCEDURE command.

The adapter supports stored procedures with input, output, and in-out parameters.

The output parameter values that are returned by stored procedures are available as result sets. These values form a single-row result set that is transferred to the client after all other result sets are returned by the invoked stored procedure. The names of the output parameters (if available) become the column titles of that result set.

Note that only the output parameters (and the returned value) referenced in the invocation string are returned to the client. As a result, users have full control over which output parameters have their values displayed.

The server supports invocation of stored procedures written according to the rules of the underlying DBMS. Note that the examples shown in this section are SQL-based. See the DBMS documentation for rules, languages, and additional programming examples.

Syntax: How to Invoke a Stored Procedure

```
SQL SQLMSS EX procname [parameter_specification1]
[,parameter_specification2]...
END
```

where:

SQLMSS

Is the ENGINE suffix for Microsoft SQL Server.

procname

Is the name of the stored procedure. It is the fully or partially qualified name of the stored procedure in the native RDBMS syntax.

You can employ either SQL or SYS naming conventions to control the separator character used for interpreting multipart names, as described in Setting Naming Conventions.

parameter_specification

IN, OUT, and INOUT parameters are supported. Use the variation required by the stored procedure:

IN

Is a literal (for example, 125, 3.14, 'abcde'). You can use reserved words as input. Unlike character literals, reserved words are not enclosed in quotation marks (for example, NULL). Input is required.

OUT

Is represented as a question mark (?). You can control whether output is passed to an application by including or omitting this parameter. If omitted, this entry will be an empty string (containing 0 characters).

INOUT

Consists of a question mark (?) for output and a literal for input, separated by a slash: /. (For example: ?/125, ?/3.14, ?/'abcde'.) The out value can be an empty string (containing 0 characters).

Example: Invoking a Stored Procedure

In this example, a user invokes a stored procedure, `edaqa.test_proc01`, supplies input values for parameters 1, 3, 5 and 7, and requests the returned value of the stored procedure, as well as output values for parameters 2 and 3.

Note that parameters 4 and 6 are omitted; the stored procedure will use their default values, as specified at the time of its creation.

```
SQL SQLMSS EX edaqa.test_proc01 125,?,?/3.14,, 'abc',, 'xyz'  
END
```

Example: Sample Stored Procedure

This stored procedure uses out and inout parameters:

```

CREATE PROCEDURE EDAQA.PROCP3 (    OUT chSQLSTATE_OUT    CHAR(5),
                                OUT intSQLCODE_OUT    INT,
                                INOUT l_name char(20),
                                INOUT f_name char(20))

    RESULT SETS 1
    LANGUAGE SQL
-----
-- SQL Stored Procedure
-----
P1: BEGIN
    -- Declare variable
    DECLARE SQLSTATE CHAR(5) DEFAULT '00000';
    DECLARE SQLCODE INT DEFAULT 0;
    -- Declare cursor
    DECLARE cursor1 CURSOR WITH RETURN FOR
        SELECT
            EDAQA.NF29005.SSN5 AS SSN5,
            EDAQA.NF29005.LAST_NAME5 AS LAST_NAME5,
            EDAQA.NF29005.FIRST_NAME5 AS FIRST_NAME5,
            EDAQA.NF29005.BIRTHDATE5 AS BIRTHDATE5,
            EDAQA.NF29005.SEX5 AS SEX5
        FROM
            EDAQA.NF29005
        WHERE
            (
                ( EDAQA.NF29005.LAST_NAME5 = l_name )
            AND
                ( EDAQA.NF29005.FIRST_NAME5 = f_name )
            );
    -- Cursor left open for client application
    OPEN cursor1;
    SET chSQLSTATE_OUT = SQLSTATE;
    SET intSQLCODE_OUT = SQLCODE;
    SET l_name = 'this is first name';
    SET f_name = 'this is last name';
END P1    @

```

Reference: Capturing Application Errors in Stored Procedures

You can capture application errors using the RAISERROR method. Any application error that is issued by the stored procedure is available in the server variable &MSSMSGTXT.

Index

&MSSMSGTXT server variable 45

A

Access File keywords
 for Microsoft SQL Server 15
 adapter configuration
 for Microsoft SQL Server 6
 adapter customization
 for Microsoft SQL Server 31
 adapter optimization
 for Microsoft SQL Server 36
 adapters
 connection attributes for Microsoft SQL Server
 7
 application requests
 for Microsoft SQL Server 5
 authentication
 for Microsoft SQL Server 6, 7
 AUTODISCONNECT command
 for Microsoft SQL Server 9

B

block size
 for Microsoft SQL Server 32

C

calling stored procedures
 for Microsoft SQL Server 43
 capturing application errors in stored procedures 45
 configuration
 for Microsoft SQL Server 6
 connection attributes
 for Microsoft SQL Server 7

connection scope
 for Microsoft SQL Server 9
 CONNECTION_ATTRIBUTES command
 for Microsoft SQL Server 7
 creating synonyms
 for Microsoft SQL Server 10
 cursor types for Microsoft SQL Server 31
 customization
 for Microsoft SQL Server 31

D

data types
 for Microsoft SQL Server 18

E

efficiency optimization
 for Microsoft SQL Server 36

I

invoking stored procedures for Microsoft SQL Server
 43

M

managing metadata
 for Microsoft SQL Server 9
 mapping data types
 for Microsoft SQL Server 18
 metadata
 for Microsoft SQL Server 9

Microsoft SQL Server Adapter 5, 6, 7, 9, 10, 15, 18, 23, 24, 31, 32, 34, 42, 43

- Access File keywords 15
- accessing remotely 6
- application requests 5
- AUTODISCONNECT command 9
- block size 32
- configuring 6
- connection attributes 7
- connection attributes on IBM i and z/OS 7
- connection attributes on UNIX 7
- connection scope 9
- CONNECTION_ATTRIBUTES command 7
- creating synonyms 10
- cursor types 31
- customizing 31
- data types 18
- DEFINE field optimization limitations 42
- explicit authentication 7
- metadata 9
- National Language Support (NLS) 23
- NLS (National Language Support) 23
- password passthru authentication 7
- preparing the environment 6
- read-only fields 24
- security 7
- stored procedures 43
- synonyms 10
- TRANSACTION command 34
- trusted authentication 7

N

- National Language Support (NLS)
 - for Microsoft SQL Server 23
- NLS (National Language Support)
 - for Microsoft SQL Server 23
- NONBLOCK mode
 - for Microsoft SQL Server 32

O

- OPTIFTHENELSE command 38, 41
- OPTIMIZATION
 - for Microsoft SQL Server 36
- OPTIMIZATION command 36
- optimizing requests 36
- optimizing requests with virtual fields 38, 41

P

- PASSRECS
 - for Microsoft SQL Server 33
- preparing the adapter environment
 - for Microsoft SQL Server Adapter 6

R

- RAISERROR method 45
- RDBMS optimization
 - for Microsoft SQL Server 36
- read-only fields for Microsoft SQL Server 24
- reporting optimization
 - for Microsoft SQL Server 36

S

- security
 - for Microsoft SQL Server 7
- SQLJOIN command 36
- stored procedures
 - for Microsoft SQL Server 43
 - specifying parameters 43
 - synonyms describing 27
- synonym creation parameters
 - for Microsoft SQL Server 11
- synonyms
 - for Microsoft SQL Server 10

T

TRANSACTION command for Microsoft SQL Server
34
transaction isolation level
for Microsoft SQL Server 35

U

Unicode data types
for Microsoft SQL Server 22
Unicode support
Microsoft SQL Server Adapter 22

