IBM SPSS Data Collection JDBC Driver User’s Guide

Note: Before using this information and the product it supports, read the general information under Notices on p. 21.

Preface

Welcome to the *IBM® SPSS® Data Collection JDBC Driver User's Guide*. This guide provides information on using the JDBC Driver.

Adobe Portable Document Format (*.pdf*) versions of the guides are available on the *IBM® SPSS® Data Collection Data Model* CD-ROM. Viewing and printing the documents requires Adobe Reader. If necessary, you can download it at no cost from *www.adobe.com*. Use the Adobe Reader online Help for answers to your questions regarding viewing and navigating the documents.

About SPSS Inc., an IBM Company

SPSS Inc., an IBM Company, is a leading global provider of predictive analytic software and solutions. The company's complete portfolio of products — data collection, statistics, modeling and deployment — captures people's attitudes and opinions, predicts outcomes of future customer interactions, and then acts on these insights by embedding analytics into business processes. SPSS Inc. solutions address interconnected business objectives across an entire organization by focusing on the convergence of analytics, IT architecture, and business processes. Commercial, government, and academic customers worldwide rely on SPSS Inc. technology as a competitive advantage in attracting, retaining, and growing customers, while reducing fraud and mitigating risk. SPSS Inc. was acquired by IBM in October 2009. For more information, visit *http://www.spss.com*.

Technical support

Technical support is available to maintenance customers. Customers may contact Technical Support for assistance in using SPSS Inc. products or for installation help for one of the supported hardware environments. To reach Technical Support, see the SPSS Inc. web site at *http://support.spss.com* or find your local office via the web site at *http://support.spss.com/default.asp?refpage=contactus.asp*. Be prepared to identify yourself, your organization, and your support agreement when requesting assistance.

Contents

Appendix

Chapter

1

IBM SPSS Data Collection JDBC Driver guide

The IBM® SPSS® Data Collection JDBC Driver allows you to read Data Collection case data in applications that support Java Database Connectivity (JDBC). The Data Collection JDBC Driver contains the following components:

- **JDBC Service Driver.** This component is only available for Windows platforms. The service driver provides a bridge between the JDBC client and the Data Collection IBM® SPSS® Data Collection Data Model component. The service driver handles data requests from the service client driver, and can be installed on the same computer or on one or more remote computers. This allows you to configure one service driver that can be used by many clients. The service driver can reduce network traffic when your case data is located on the same computer where the service driver is installed (all queries occur on the same server). Only the resulting cases are sent to the service client. There may also be performance improvements when the server has a faster processor or more RAM than the service client machines.
- **Client Service Driver.** This component is available for Windows, UNIX, and Linux operating systems. The service client driver provides an interface between the client application, that needs to read the Data Collection case data, and the service driver that handles the data request.

IBM SPSS Data Collection JDBC Driver system requirements

The minimum system requirements for the IBM® SPSS® Data Collection JDBC Driver are:

JDBC Service Driver

- Microsoft® Windows® XP Professional (x86 32-bit edition) with Service Pack 3, Microsoft Windows Server 2003 or 2008 Standard or Enterprise Edition (x86 32-bit or x64 64-bit edition) with Service Pack 2, Microsoft Windows Vista Business or Enterprise Edition (x86 32-bit or x64 64-bit edition), or Microsoft Windows 7 Professional or Enterprise Edition (32-bit or x64 64-bit edition)
- Approximately 100MB of available hard disk space.
- Data Collection IBM® SPSS® Data Collection Data Model 6 must be installed prior to installing the JDBC service driver.

Client Service Driver

Windows

- Microsoft® Windows® XP Professional (x86 32-bit edition) with Service Pack 3, Microsoft Windows Server 2003 or 2008 Standard or Enterprise Edition (x86 32-bit or x64 64-bit edition) with Service Pack 2, Microsoft Windows Vista Business or Enterprise Edition (x86 32-bit or x64 64-bit edition), or Microsoft Windows 7 Professional or Enterprise Edition (32-bit or x64 64-bit edition)

Linux

- Linux®

Sun Solaris

- Sun Solaris 9 (2.9) (32-bit or 64-bit SPARC), Sun Solaris 10 (32-bit or 64-bit SPARC)

HP-UX

- HP-UX 11 (64-bit Itanium)

AIX

- AIX®

J2SE 1.4.2 (or higher)

- Must be installed prior to installing the Client service driver.

Installation instructions

Installing the JDBC service driver

1. Navigate to the *\JDBC Driver\Server* directory on the IBM® SPSS® Data Collection Data Model installation CD.
2. Double-click the file *IBM SPSS Data Collection JDBC Service Driver.exe*.
3. This starts the IBM® SPSS® Data Collection JDBC Service Driver Setup wizard. Click Next.
4. The License Agreement page displays the software license agreement. If you accept the agreement, click I accept the terms in the license agreement. Click Next to continue.
5. The Service Configuration page displays the system port number that the JDBC service driver will use. Enter an appropriate port number, or keep the default setting of **19999**, and click Next.

   Important: You must change the default setting if port 19999 is in use by another application.
6. The Destination Folder page displays the path where the JDBC service driver will be installed. You can change the path or keep the default location. Click Next to continue.
7. The Ready to Install the Program page displays the selected installation options. Click Next to begin the installation.
8. After the installation is complete, click Finish.

Note: The JDBC service driver requires the file *mrOledb.dll* to retrieve data from Data Collection applications. It is recommended that you install the Data Collection IBM® SPSS® Data Collection Data Model 6 before installing the driver.

Uninstalling the JDBC service driver

1. Open the Windows Control Panel and double-click the Add or Remove Programs icon.
2. Select Data Collection JDBC Service Driver, and click Remove.
3. Click Yes on the Add or Remove programs warning popup.

Installing the client service driver

1. Navigate to the *\JDBC Driver\Client* directory on the installation CD.
   - **PHjc.jar** - the driver JAR file that is used directly in Java applications. The driver class is: `com.spss.datacollection.jdbc.openaccess.OpenAccessDriver`
   - **dcjcinstall.package -** the installation package for IBM® SPSS® Collaboration and Deployment Services Package Manager. You can install the driver as a IBM SPSS Collaboration and Deployment Services component.

Installing the client service driver for BIRT Report Designer for IBM SPSS

1. Navigate to the *\JDBC Driver\Client* directory on the Data Model installation CD.
2. Double-click the file *IBM SPSS Data Collection JDBC Client Service Driver for BIRT Report Designer.exe*. This starts the Client Service Driver for BIRT Report Designer for IBM® SPSS® Setup wizard. Click Next to continue.
3. The License Agreement step displays the software license agreement. Select I accept the terms in the license agreement and click Next to continue.
4. The Destination Folder step displays the path where the client service driver will be installed. By default, this path is the default BIRT Report Designer for IBM SPSS 4.0 installation directory. If BIRT Report Designer for IBM SPSS is installed in another directory, or the version is not 4.0, change this path to match the correct BIRT Report Designer for IBM SPSS path. Click Next to continue.
5. Click Next on the Ready to Install the Program step to begin the installation.
6. click Finish. After the installation is complete.

Connection string

There are a number of connection properties that need to be specified when you access data using the JDBC Driver. When you access data using the IBM® SPSS® Data Collection products, this is generally invisible, because the products automatically set up all of the connection properties for you. However, when you use the JDBC Driver to access data, the connection properties need to be defined.

```
<connection-string> ::= jdbc:paswdatacollection://<hostname>:<port>; CustomProperties=(<properties>)
<properties> ::= <property>; | <property>; <properties>
<property> ::= <property-keyword> = ["]<property-value>["]
<property-keyword> ::= <identifier>
<property-value> ::= character-string
```

Example

```
jdbc:paswdatacollection://dc-server:19999;CustomProperties=(Data Source=mrXmlDsc;Location=household.xml;
Initial Catalog=[INSTALL_FOLDER]\IBM\SPSS\DataCollection\6\DDL\Data\XML\household.mdd)
```

Refer to the *Connection Properties* and *Connecting to the IBM® SPSS® Data Collection Data Model* topics in the IBM® SPSS® Data Collection Developer Library for more information regarding connection properties.

Standard connection properties

Property Name	**Description**	**Data Type**
Connect Timeout	The length of time, in seconds, to wait for the initialization to complete. (not required)	A 32-bit, signed integer.
Data Source	The CDSC name to be used with the connection. This is the component name of the CDSC *ProgID* (for example, *mrRdbDsc*). The JDBC driver can automatically selects the CDSC. (not required)	String
Initial Catalog	The name or location of the metadata source. For file or directory-based data stores, this is the path to the file or directory. The provider will treat a relative path as being relative to the metadata file specified in the Initial Catalog property. If Initial Catalog is not set, the provider will treat a relative path as being relative to the current directory. The provider resolves a relative path to an absolute path before passing it to the CDSC. For OLE DB-based data stores, this is the OLE DB connection string. See the *Connecting to a Relational MR Database* topic in the Data Collection Developer Library for more information. (required)	String
Location	The name or location of the data source. For file or directory-based data stores, this is the path to the file or directory. (not required)	String
Mode	This is used internally to filter the CDSCs in the Data Link Properties dialog box (for example, so that only read-enabled CDSCs are listed when the Mode setting is **read**). Other uses of this property are reserved for future use. (not required)	String
Password	The password to connect to the data source. This property should not be a blank string. (required)	String
User ID	The user ID to connect to the data source. The user ID should have the appropriate authority to logon to the workstation where the service driver is installed. (required)	String

Note: Connection string property values cannot include parenthesis characters ( ).

Updating the JDBC Service Driver to support custom connection property prefixes

By default, custom connection properties that are prefixed by *MR* or *MR Init* are not supported in the connection string. If you want to specify these custom connection properties they must be manually declared in the JDBC Service Driver's configuration:

1. After installing the JDBC Service Driver, go to the Windows Control Panel > Administrative Tools > Services.

2. In the Services list, locate and select the service DataCollectionJDBCDriverService.

3. Select Stop from the Action menu.

4. Navigate to the \cfg directory in the JDBC Service Driver installation path (default path is: [INSTALL_FOLDER]\IBM\SPSS\DataCollection\6\JDBC Server).

5. Open the file *oadm.ini* with a text editor (Notepad for example).

6. Search for the keyword *DataSourceIPCustomProperties*; the default value is:

   ```
   User ID=;Password=;Data Source=;Initial Catalog=;Location=;Mode=Read;Connect Timeout=;
   ```

7. Append the appropriate property name at the end of the value. For example, if you want to add the property MR Init Category Names, the value should be written as:

   ```
   User ID=;Password=;Data Source=;Initial Catalog=;Location=;Mode=Read;Connect Timeout=;MR Init Category Names=;
   ```

8. Save the *oadm.ini* file.

9. Go back to the Services dialog.

10. In the Services list, locate and select the service DataCollectionJDBCDriverService.

11. Select Start from the Action menu and close the Services dialog.

IBM SPSS Data Collection JDBC Driver specification

The section outlines the IBM® SPSS® Data Collection JDBC Driver specification, and is composed of the following topics:

- SQL syntax
- Database metadata
- IBM SPSS Data Collection data types
- Query from VDATA
- Query from HDATA
- Return category value
- Return level field

SQL syntax

Tables

The IBM® SPSS® Data Collection JDBC Driver supports the VDATA, HDATA and SYS.* tables.

VDATA and HDATA are supported by the Data Collection OLE DB provider. After a query, the JDBC driver returns a standard JDBC result set (the JDBC driver converts results returned by the Data Collection OLE DB provider into a JDBC result set). The transformation can result in multiple tables being returned. Refer to Query from VDATA and Query from HDATA for more information.

- **VDATA** – a single, flat table that contains a column for each variable that can be presented in a flattened form.
- **HDATA** – a set of hierarchical tables (the top-level table is called HDATA). Each lower-level table represents the data in a loop or grid.

Refer to the *Virtual Tables* topic in the IBM® SPSS® Data Collection Developer Library for more information.

SQL Syntax

The JDBC driver works in pass-through mode. Refer to the *SQL Queries* section in the Data Collection Developer Library for more detailed information regarding SQL query statements. The section provides topics for the following:

- Basic SQL Queries
- Advanced SQL Queries
- Hierarchical SQL Queries
- Advanced Hierarchical SQL Queries

Expression Evaluation

The Data Collection Evaluate component implements expression parsing and evaluation. The component is used by the Case Data Model (CDM), the Metadata Model (MDM), and the mrScript engine. The Evaluate component supports standard expression grammar. The parser is unique, however, in its support for categorical variables and its optional support for single or double-quoted string literals. Expression evaluation is documented in detail in the *Expression Evaluation* section in the Data Collection Developer Library. The section provides links to the following topics:

- **Operator Precedence.** Describes the order in which operators are evaluated when there are multiple operators in an expression.
- **Arithmetic Operators.** Describes the behavior of the addition (+), subtraction (-), multiplication (*), division (/), and modulus (Mod) arithmetic operators on Data Model variables of various types.
- **Categorical Set Logic.** Summarizes the set operations for categorical variables.

- **Comparison Operators.** Describes the behavior of the equal to (=), not equal to (<>), less than (<), less than or equal to (<=), greater than (>), and greater than or equal to (>=) comparison operators on Data Model variables of various types.
- **Categorical Comparison Operators.** Summarizes how comparison operators work with categorical variables.
- **Like Operator.** Describes the Like operator, which is a special comparison operator that can be used to compare two text values.
- **Logical Operators.** Describes the And, Or, Xor, and Not logical operators.
- **Complex Expressions.** Provides some examples of using more than one operator in an expression.
- **Null Values.** Describes how the Evaluate component handles NULL values.

IBM SPSS Data Collection Function Library

The Data Collection Function Library supplements basic expression evaluation with more advanced operations. The Function Library is documented in detail in the *Data Collection Function Library* section in the Data Collection Developer Library.

The functions are divided into the following groups:

- **Categorical Functions.** Functions that operate on categorical data only.
- **Text, Categorical, or Array Functions.** Functions that operate on text or categorical data, or arrays.
- **Text Functions.** Functions that operate on text data only.
- **Date and Time Functions.** Functions that operate on date and time data only.
- **Conversion Functions.** Functions that convert data from one data type to another.
- **Random Number Functions.** Functions for generating random numbers.
- **List Functions.** Functions for ordering, and retrieving items from, category lists, arrays, and collections of objects.
- **Mathematical Functions.** A number of mathematical functions.
- **Miscellaneous Functions.** Functions that do not apply to any of the other groups.

Aggregate functions

The Case Data Model provides support for a number of aggregate functions. These functions can be used in SQL queries to summarize information from multiple rows. Supported aggregate functions are shown in the following table.

Aggregate Function	Description
AVG(expr)	Returns the average of the values defined in the expression. This function can be used with numeric data only.
BASE(expr)	Returns the total number of cases included in the expression. Generally, the base includes every case for which the value is not NULL. However, if the IsInBase attribute has been set to True, the case is included in the base regardless of the value. Conversely, if the IsInBase attribute has been set to False, the case is excluded from the base regardless of the value.

Aggregate Function	Description
COUNT(expr)	Returns a count of the cases in a column.
SUM(expr)	When used with numeric data, returns the sum of the values. When used with categorical data, it returns the union of the categorical values.
MIN(expr)	Returns the lowest value defined in the expression.
MAX(expr)	Returns the highest value defined in the expression.
STDEV(expr)	Returns the standard deviation of the values defined in the expression. This function can be used with numeric data only. The standard deviation is a measure of dispersion around the mean. In a normal distribution, 68% of cases fall within one standard deviation of the mean and 95% of cases fall within 2 standard deviations. For example, if the mean age is 45 with a standard deviation of 10, then 95% of the cases would be between 25 and 65 in a normal distribution.

Database metadata

The IBM® SPSS® Data Collection JDBC driver supports the following metadata:

getColumns

Retrieves a description of table columns available in the specified catalog. Only column descriptions matching the catalog, schema, table and column name criteria are returned. The results are ordered by TABLE_CAT, TABLE_SCHEM, TABLE_NAME, and ORDINAL_POSITION.

```
ResultSet getColumns(String catalog,
            String schemaPattern,
            String tableNamePattern,
            String columnNamePattern)
            throws SQLException
```

The JDBC driver ignores all parameters and returns all columns. The driver returns all fields in VDATA, HDATA, array objects, and grid objects. The field name can be used in SQL.

getSchemas

Retrieves the available database schema names. The results are ordered by TABLE_CATALOG and TABLE_SCHEM.

```
ResultSet getSchemas()
            throws SQLException
```

getTables

Retrieves a description of the tables available in the given catalog. Only table descriptions matching the catalog, schema, table name, and type criteria are returned. The results ordered by TABLE_TYPE, TABLE_CAT, TABLE_SCHEM, and TABLE_NAME.

```
ResultSet getTables(String catalog,
            String schemaPattern,
            String tableNamePattern,
            String[] types)
            throws SQLException
```

The JDBC driver ignores all parameters and returns all tables.

getTypeInfo

Retrieves a description of all the supported database data types. The results are ordered by DATA_TYPE and then by how closely the data type maps to the corresponding JDBC SQL type.

If the database supports SQL distinct types, then getTypeInfo() will return a single row with a TYPE_NAME of DISTINCT and a DATA_TYPE of TYPES.DISTINCT. If the database supports SQL structured types, then getTypeInfo() returns a single row with a TYPE_NAME of STRUCT and a DATA_TYPE of Types.STRUCT. If SQL distinct or structured types are supported, then information on the individual types can be obtained from the getUDTs() method.

Data Collection TYPE_NAME	**Data Collection Data Type**	**JDBC Metadata DATA_TYPE**	**JDBC Metadata TYPE_NAME**	**Java Class**
DOUBLE	5	8	double	java.lang.Double
LONG	3	4	integer	java.lang.Integer
DATE	7	93	timestamp	java.sql.Timestamp
BOOLEAN	11	-7	boolean	java.lang.Boolean
TEXT	130	-1	longvarchar	java.lang.String
CATEGORICAL	129	12	varchar	java.lang.String
OBJECT	128	Ignore		
LEVEL	136	-2	BINARY	byte[]

IBM SPSS Data Collection data types

Refer to the sections in the IBM® SPSS® Data Collection Developer Library for more information:

- Data Types
- MDM Document Structure
- Variable Type Overview

Query from VDATA

When querying data from VDATA, the JDBC driver converts data from the IBM® SPSS® Data Collection OLE DB provider to the JAVA result set using the following rules:

Data Collection Data Type	**Description**	**JAVA Result Set**
Double	A 64-bit floating-point number with at least 15 digits of precision.	A single column: Double

Data Collection Data Type	**Description**	**JAVA Result Set**
Long	A 32-bit signed integer.	A single column: Integer
Text	A character string. The maximum length of a text variable is dependent on the DSC. The string may or may not support Unicode, depending on how the CDSC has implemented its storage. The Provider implements all strings using Unicode.	A single column: String
Categorical	A single or multiple categorical response type. The maximum number of responses available varies between DSCs. A categorical response is implemented in the Provider as a string. The responses are formatted within curly braces, ({, }), with each response delimited by a comma. For example, {2,3,8} represents the responses 2, 3, and 8. Depending on how the data was collected, the order of responses in the string can be the order in which they were chosen in response to the question. This can be important for analysis.	Column.Type is Categorical 129. The type is returned as a getTypeInfo method of DatabaseMetadata. The return value is determined by the connection properties. For more information, see the topic Return category value on p. 17.
Date	A 64-bit OLE date. The OLE date type contains both date and time and it is represented using the DATE type.	A single column: DATE
Object	Not implemented.	N/A
Boolean	A Boolean value of either True or False. This is implemented internally using VARIANT_BOOL. However, when a Boolean is converted to a Long or a Double, it will be converted to 1 for True and 0 for False.	A single column: Boolean
Level	A hierarchical value. A level contains a child recordset.	N/A for VDATA; a single BINARY column for HDATA
Array	Also called a Loop. These represent loops, grids, and levels. The Array object defines a question or set of questions that are to be asked more than once.	Multiple columns. Refer to the *Query Array field in VDATA* or *Query Array field in HDATA* section for more information.
Class	These represent a grid question. A Grid object can be considered a special case of the Array object.	Multiple columns. Refer to the *Query Class field in VDATA* or *Query Class field in HDATA*section for more information.

Data Collection Data Type	Description	JAVA Result Set
Compound	These group categorical and grid questions that share a category list, typically for presentation in a paper questionnaire.	Multiple columns. Refer to the *Query Compound field in VDATA* or *Query Compound field in HDATA* section for more information.
Grid	These are used to group questions.	Multiple columns. Refer to the *Query Grid field in VDATA* or *Query Grid field in HDATA*section for more information.

Query array field in VDATA

If an array is unbound, it is not possible to query data from VDATA, but is it is possible to query flattened data from HDATA. You cannot query an array field such as `SELECT person FROM VDATA`. You can however query sub-fields in an array field because although array fields are not columns in VDATA, sub-fields are treated as columns. If the array field is a bound array, the result set is converted into multiple columns, such as:

```
ArrayField[M].subfield, ArrayField[M+1].subfield, ..., ArrayField[N].subfield
```

M is the lower-bound value and N is upper-bound value.

Example

In the following example (based on data from the IBM® SPSS® Data Collection Developer Library *household.mdd* example), the *person* field is an array that includes the *age* sub-field. The *person* field is bound from 1 to 6. To query the *age* field data, the query statement would be:

```
SELECT person[1].age
, person[2].age
, person[3].age
, person[4].age
, person[5].age
, person[6].age
 FROM VDATA
```

The result data is:

person[1].age	person[2].age	person[3].age	person[4].age	person[5].age	person[6].age
25	(null)	(null)	(null)	(null)	(null)
45	43	15	12	(null)	(null)
72	81	(null)	(null)	(null)	(null)
32	9	2	(null)	(null)	(null)
29	31	(null)	(null)	(null)	(null)
63	(null)	(null)	(null)	(null)	(null)
38	39	8	7	4	(null)
25	23	26	(null)	(null)	(null)
42	7	(null)	(null)	(null)	(null)

Query class field in VDATA

A class field is a block of sub-fields. You cannot query a class field such as SELECT Respondent FROM VDATA. You can however query sub-fields in a class field because although class fields are not columns in VDATA, sub-fields are treated as columns.

Example

In the following example (based on data from the Data Collection Developer Library *museum.mdd* example), the *Respondent* field is a class that includes the *Serial* and *Origin* sub-fields. The *person* field is bound from 1 to 6. To query the *Serial* and *Origin* field data, the query statement would be:

```
SELECT Respondent.Serial
, Respondent.Origin
 FROM VDATA
```

The result data is:

Respondent.Serial	Respondent.Origin
1	(null)
2	(null)
3	(null)

Query grid field in VDATA

You cannot query a grid field such as SELECT rating FROM VDATA. You can however query sub-fields in a grid field because although grid fields are not columns in VDATA, sub-fields are treated as columns. There are typically three kinds of grids: single response grids, multiple responses grids, and numeric grids.

Single response grids and multiple responses grids

You can use gridName[{elementName}].gridSubField to obtain a sub-field's value. Refer to Return category value for more information returning categorical values in JDBC.

Example

In the following example (based on data from the Data Collection Developer Library *museum.mdd* example), *rating* is a multiple response grid:

```
SELECT
 rating[{Dinosaurs}].Column
,  rating[{Conservation}].Column
, rating[{Fish_and_reptiles}].Column
FROM VDATA
```

The result data is:

rating[{Dinosaurs}].Column	rating[{Conservation}].Column	rating[{Fish_and_reptiles}].Column
{52}	{51}	{52}
{51}	{}	{50}
{48}	{}	{51}
{51}	{51}	{51}

Numeric grids

You can use gridName [{elementName}].numericName to obtain a sub-field's value.

Example

Using data from the Data Collection Developer Library *short_drinks.mdd* example:

```
SELECT
numdrksz[{TEA}].mond
, numdrksz[{COFFEE}].mond
, numdrksz[{TEA}].tuesd
, numdrksz[{COFFEE}].tuesd
FROM VDATA
```

The result data is:

numdrksz[{TEA}].mond	numdrksz[{COFFEE}].mond	numdrksz[{TEA}].tuesd	numdrksz[{COFFEE}].tuesd
5	2	4	3
3	4	3	3
6	0	6	0
0	3	0	2

Query compound field in VDATA

A compound field is a class that includes multiple grid fields. All grids share a same set of categorical elements that are stored in the compound field. You cannot query a compound field such as SELECT plans FROM VDATA. Compound fields are not columns in VDATA.

You can use compoundName.gridName[{elementName}].gridSubFieldName to obtain a sub field's value.

Summary

VDATA is a flat view; all columns can be obtained via the getColumns method. The negative aspects of using VDATA are:

- Bound array fields and grid fields are presented in expanded mode.
- You cannot query unbound array data.

Query from HDATA

When you query data from HDATA, the IBM® SPSS® Data Collection JDBC Driver converts data from the Data Collection OLE DB Provider to the JAVA result set using the following rules:

Data Collection Data Type	**Description**	**JAVA Result Set**
Double	A 64-bit floating-point number with at least 15 digits of precision.	A single column: Double
Long	A 32-bit signed integer.	A single column: Integer
Text	A character string. The maximum length of a text variable is dependent on the DSC. The string may or may not support Unicode, depending on how the CDSC has implemented its storage. The Provider implements all strings using Unicode.	A single column: String
Categorical	A single or multiple categorical response type. The maximum number of responses available varies between DSCs. A categorical response is implemented in the Provider as a string. The responses are formatted within curly braces, ({, }), with each response delimited by a comma. For example, {2,3,8} represents the responses 2, 3, and 8. Depending on how the data was collected, the order of responses in the string can be the order in which they were chosen in response to the question. This can be important for analysis.	Column.Type is Categorical 129. The type is returned as a getTypeInfo method of DatabaseMetadata. The return value is determined by the connection properties. For more information, see the topic Return category value on p. 17.
Date	A 64-bit OLE date. The OLE date type contains both date and time and it is represented using the DATE type.	A single column: DATE
Object	Not implemented.	N/A
Boolean	A Boolean value of either True or False. This is implemented internally using VARIANT_BOOL. However, when a Boolean is converted to a Long or a Double, it will be converted to 1 for True and 0 for False.	A single column: Boolean
Level	A hierarchical value. A level contains a child recordset.	N/A for VDATA; a single BINARY column for HDATA
Array	Also called a Loop. These represent loops, grids, and levels. The Array object defines a question or set of questions that are to be asked more than once.	Multiple columns. Refer to the *Query Array field in VDATA* or *Query Array field in HDATA* section for more information.

Data Collection Data Type	Description	JAVA Result Set
Class	These represent a grid question. A Grid object can be considered a special case of the Array object.	Multiple columns. Refer to the *Query Class field in VDATA* or *Query Class field in HDATA*section for more information.
Compound	These group categorical and grid questions that share a category list, typically for presentation in a paper questionnaire.	Multiple columns. Refer to the *Query Compound field in VDATA* or *Query Compound field in HDATA* section for more information.
Grid	These are used to group questions.	Multiple columns. Refer to the *Query Grid field in VDATA* or *Query Grid field in HDATA*section for more information.

Query array field in HDATA

In HDATA, array fields are considered level. Querying a level field is same as querying all sub-fields of the level field. For more information, see the topic Return level field on p. 18.

You can query array object sub-fields from the array table. Refer to the example below for more information.

Array objects contain the specialty field LevelID, which is a sequential number in the current level.

Example

Using data from the IBM® SPSS® Data Collection Developer Library *household.mdd* example:

```
SELECT
 ^.housetype
, ^.address
, levelid
, age
, person
FROM HDATA.person
```

The result data is:

^.housetype	^.address	levelid	age	person
{59}	15B Park Avenue, Harrogate, HG1 4TY	1	25	1
{61}	46 Freedom Lane, Brighton, BN2 3YT	1	45	2
{61}	46 Freedom Lane, Brighton, BN2 3YT	2	43	3
{61}	46 Freedom Lane, Brighton, BN2 3YT	3	15	4
{61}	46 Freedom Lane, Brighton, BN2 3YT	4	12	5
{60}	The Meadows, Clifton Heights, BS34 3EG	1	72	6
{60}	The Meadows, Clifton Heights, BS34 3EG	2	81	7
{62}	23 Steep Hill, Norfold, CB64 5TY	1	32	8
{62}	23 Steep Hill, Norfold, CB64 5TY	2	9	9

^.housetype	^.address	levelid	age	person
{62}	23 Steep Hill, Norfold, CB64 5TY	3	2	10
{61}		1	29	11
{61}		2	31	12
...	...	...	...	...

Query grid field in HDATA

In HDATA, grid fields are considered level. Querying a level field is same as querying all sub-fields of the level field. For more information, see the topic Return level field on p. 18.

You can query grid object sub-fields from the array table. Refer to the examples below for more information.

Grid objects contain the specialty field LevelID, which is an element value of the current grid.

Single response grids and multiple responses grids

Refer to the topic Return category value for information on returning categorical values in JDBC.

Example

Using data from the Data Collection Developer Library *museum.mdd* example:

```
SELECT
^.name
,levelid
,column
FROM HDATA.RATING
```

The result data is:

^.name	levelid	column
MyName	{31}	{52}
MyName	{32}	{}
MyName	{33}	{}
MyName	{34}	{}
MyName	{35}	{}
MyName	{36}	{}
MyName	{37}	{}
MyName	{38}	{51}
MyName	{39}	{52}
MyName	{40}	{}
MyName	{41}	{50}
MyName	{42}	{51}
MyName	{43}	{}
MyName	{44}	{51}
MyName	{45}	{}

^.name	levelid	column
MyName	{18}	{}
	...	...

Numeric grids

Example

Using data from the Data Collection Developer Library *short_drinks.mdd* example:

```
SELECT
 Levelid
,mond
,tuesd
FROM HDATA.numdrksz
```

The result data is:

LevelID	mond	tuesd
{48}	5	4
{49}	2	3
{50}	0	0
{51}	0	0
{52}	0	1
{53}	0	0
{54}	0	0
{48}	3	3
{49}	4	3
{50}	0	0
{51}	0	0
{52}	0	0
{53}	0	0
{54}	0	0
...	...	

Return category value

Single Response Category

Return category as a string, such as {100}.

Multiple Response Category

Return category as a string, such as {100,102}.

Return level field

In HDATA, array fields and grids field are considered level (hierarchical table).

If the query statement is SELECT * FROM HDATA, the current level is HDATA. If the query statement is SELECT * FROM HDATA.person, the current level is HDATA.person.

In the IBM® SPSS® Data Collection OLE DB Provider, when querying fields are the same as the current level (or upper, then current level), the field's result should be considered as a normal field; otherwise querying fields are returned as a sub-table (level object).

Example: query content level fields and parent level fields

Using data from the IBM® SPSS® Data Collection Developer Library *household.mdd* example:

```
SELECT
 ^.housetype
, ^.address
, levelid
, age
, person
FROM HDATA.person
```

The result data is:

^.housetype	^.address	levelid	age	person
{59}	15B Park Avenue, Harrogate, HG1 4TY	1	25	1
{61}	46 Freedom Lane, Brighton, BN2 3YT	1	45	2
{61}	46 Freedom Lane, Brighton, BN2 3YT	2	43	3
{61}	46 Freedom Lane, Brighton, BN2 3YT	3	15	4
{61}	46 Freedom Lane, Brighton, BN2 3YT	4	12	5
{60}	The Meadows, Clifton Heights, BS34 3EG	1	72	6
{60}	The Meadows, Clifton Heights, BS34 3EG	2	81	7
{62}	23 Steep Hill, Norfold, CB64 5TY	1	32	8
{62}	23 Steep Hill, Norfold, CB64 5TY	2	9	9
{62}	23 Steep Hill, Norfold, CB64 5TY	3	2	10
{61}		1	29	11
{61}		2	31	12
...	...	...	...	...

housetype	Address	levelid	age	person
{59}	15B Park Avenue, Harrogate, HG1 4TY	1	25	1
{61}	46 Freedom Lane, Brighton, BN2 3YT	1	45	2
		2	43	3
		3	15	4
		4	12	5
{60}	The Meadows, Clifton Heights, BS34 3EG	1	72	6
		2	81	7

housetype	Address	levelid	age	person
{62}	23 Steep Hill, Norfold, CB64 5TY	1	32	8
		2	9	9
		3	2	10
{61}		1	29	11
		2	31	12
...	...	...	...	...

Return level object

Since the JDBC Driver only supports basic JDBC types, when the driver receives a level object from the Data Collection OLE DB Provider, the driver transports the level object as a binary array. At the Java client site, users are able to identify column values as a level of a sub-table by using the data type. If you want to place data inside a sub-table, you must query data directly from the table (for example, `SELECT * FROM HDATA.person`).

Troubleshooting

After installing the JDBC Service Driver, how can I verify that it is functioning properly? Is there a log that can help debug potential problems?

There is a Windows service named *DataCollectionJDBCDriverService* that runs even when the Data Model is not installed on your machine. When problems occur, you can check the *DataCollectionJDBCDriverService* service log in the Windows Event Viewer. You can access the Event Viewer from the Windows Control Panel:

Start > Control Panel > Administrative Tools > Event Viewer

What permissions are required to use the JDBC Client Driver?

The account referenced in the connection string should be a Windows user, and must have permission to log onto the JDBC Service Driver machine. Note that blank passwords are not accepted.

How can I convert category values, such as {25,26}, to readable strings via the JDBC Client Driver?

By default, the JDBC Client Driver returns category values for the type `CATEGORICAL`. You can use the method `FormatValue()` to convert the values to a label. For example, if you want to convert a column named `location` in an HDATA table, you can write a query such as:

```
SELECT FormatValue(location, location, 'b') from HDATA
```

Note that the first `location` in the above query is the variable name defined in the metadata and the second `location` is the column name in HDATA. Refer to the topic “FormatValue” in the *IBM® SPSS® Data Collection Developer Library* for more information.

How to obtain the category element label

Use the metadata mechanisms and, if required, use the *DefinedCategories* function. Metadata should be supplied via the standard JDBC metadata mechanism, not new SQL syntax. SQL syntax should be used as a pass-through to mrOleDB.

Example for multi-response categoricals

```
// Categorical string (unique values):
SELECT Museums FROM VDATA

// Categorical string (category names):
SELECT Museums.Format('a') FROM VDATA

// Categorical string (category labels):
SELECT Museums.Format('b') FROM VDATA

// Category set (unique values)
SELECT Museums.GetAnswer(0), Museums.GetAnswer(1), Museums.GetAnswer(2), Museums.GetAnswer(3), Museums.GetAnswer(4) FR

// Category set (category names)
SELECT Museums.GetAnswer(0).Format('a'), Museums.GetAnswer(1).Format('a'), Museums.GetAnswer(2).Format('a'), Museums.GetAns

// Dichotomy set
SELECT Museums.HasAnswer({national_museum_of_science}) As national_museum_of_science,
Museums.HasAnswer({museum_of_design}) As museum_of_design,
Museums.HasAnswer({institute_of_textiles_and_fashion}) As institute_of_textiles_and_fashion,
Museums.HasAnswer({archeological_museum}) As archeological_museum,
Museums.HasAnswer({national_art_gallery}) As national_art_gallery,
Museums.HasAnswer({northern_gallery}) As northern_gallery,
Museums.HasAnswer({other}) As other,
Museums.HasAnswer({not_answered }) As not_answered FROM VDATA
```

The labels for all elements can be returned from mrOleDB. For example: `SELECT Museums.DefinedCategories().Format('b') FROM VDATA`

Appendix A

Notices

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: SPSS INC., AN IBM COMPANY, PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. SPSS Inc. may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-SPSS and non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this SPSS Inc. product and use of those Web sites is at your own risk.

When you send information to IBM or SPSS, you grant IBM and SPSS a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Information concerning non-SPSS products was obtained from the suppliers of those products, their published announcements or other publicly available sources. SPSS has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-SPSS products. Questions on the capabilities of non-SPSS products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to SPSS Inc., for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. SPSS Inc., therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided “AS IS”, without warranty of any kind. SPSS Inc. shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks of IBM Corporation, registered in many jurisdictions worldwide. A current list of IBM trademarks is available on the Web at *http://www.ibm.com/legal/copytrade.shmtl*.

SPSS is a trademark of SPSS Inc., an IBM Company, registered in many jurisdictions worldwide.

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

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other product and service names might be trademarks of IBM, SPSS, or other companies.

Adobe product screenshot(s) reprinted with permission from Adobe Systems Incorporated.

Microsoft product screenshot(s) reprinted with permission from Microsoft Corporation.

Index