MDCStoreUtils API Reference Guide

MDCStore™ Database 2.0 or later

Microsoft SQL Server and Oracle

5000957 C
August 2012

Molecular Devices

This document is provided to customers who have purchased Molecular Devices, LLC ("Molecular Devices") equipment, software, reagents, and consumables to use in the operation of such Molecular Devices equipment, software, reagents, and consumables. This document is copyright protected and any reproduction of this document, in whole or any part, is strictly prohibited, except as Molecular Devices may authorize in writing.

Software that may be described in this document is furnished under a license agreement. It is against the law to copy, modify, or distribute the software on any medium, except as specifically allowed in the license agreement. Furthermore, the license agreement may prohibit the software from being disassembled, reverse engineered, or decompiled for any purpose.

Portions of this document may make reference to other manufacturers and/or their products, which may contain parts whose names are registered as trademarks and/or function as trademarks of their respective owners. Any such usage is intended only to designate those manufacturers' products as supplied by Molecular Devices for incorporation into its equipment and does not imply any right and/or license to use or permit others to use such manufacturers' and/or their product names as trademarks.

Molecular Devices makes no warranties or representations as to the fitness of this equipment for any particular purpose and assumes no responsibility or contingent liability, including indirect or consequential damages, for any use to which the purchaser may put the equipment described herein, or for any adverse circumstances arising therefrom.

For research use only. Not for use in diagnostic procedures.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

The trademarks mentioned herein are the property of Molecular Devices, LLC or their respective owners. These trademarks may not be used in any type of promotion or advertising without the prior written permission of Molecular Devices, LLC.

Patents: http://www.moleculardevices.com/productpatents

Product manufactured by Molecular Devices, LLC.

1311 Orleans Drive, Sunnyvale, California, United States of America 94089.

Molecular Devices, LLC is ISO 9001 registered.

© 2012 Molecular Devices, LLC.

All rights reserved.

Contents

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Preface

The preface contains the following sections:

Who This Manual Is For

This manual is written for those who wish to build an application that accesses and exchanges data with an MDCStore™ database. Use of the MDCStore API is only available under the MDCStore API limited licensing agreement.

About this Manual

This manual contains the following chapters:

Title	Overview
Chapter 1 Introduction to the MDCStoreUtils API	Provides an introduction the MDCStore Utilities API
Chapter 2 Database Connection Functions	Contains functions that can be used with database connections
Chapter 3 Results Set Functions	Contains functions that can be used with results sets
Chapter 4 Result Set Data Types Functions	Contains functions that can be used with results set data types
Chapter 5 Error Handling Functions	Contains error handing functions
Chapter 6 BLOB Functions	Contains functions that can be used with BLOBs
Chapter 7 Measurement Sets and Results Functions	Contains functions that can be used with measurement sets and results
Chapter 8 Common Database Functions	Contains common database functions
Chapter 9 Plate Functions	Contains functions that can be used with plates
Chapter 10 Acquisitions Functions	Contains functions that can be used with acquisitions
Chapter 11 Security Functions	Contains security functions
Chapter 12 Datasets Functions	Contains functions that can be used with datasets
Chapter 13 Assay Images and Assay Normalization Functions	Contains functions that can be used with assay images and assay normalization
Chapter 14 Quicklist Functions	Contains quicklist functions
Chapter 15 Application Data Table Functions	Contains functions that can be used with acquisition data table functions

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Title	Overview
Chapter 16 File Data and Macro Functions	Contains functions that can be used with file data and macros
Chapter 17 Structures, ENUMS, and Definitions	Describes data structures enums, definitions and error codes
Chapter 18 Virtual Callback Classes	Describes virtual call back classes
Chapter 19 Usage Examples	Contains examples of how to use the MDCStore Utilities API

Conventions

Within the scope of this manual, the following typographical conventions are used.

WARNING! A warning indicates an operation that may cause personal injury if precautions are not followed.

CAUTION! Indicates an operation that may cause damage to the instrument, device, or data, if the precautions are not followed.

Note: Provides essential information for the completion of a procedure.

Tip! Provides useful information that helps apply the techniques and procedures in the text to your specific needs, and provides shortcuts, but is not essential to the completion of a procedure.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Introduction to the MDCStoreUtils API

1

What is the MDCStoreUtils API?

The MDCStoreUtils API is an open application programming interface (API) created using ODBC classes to work with the MDCStore™ database. It provides a set of functions to allow MDC customers to build applications that access and exchange data with the MDCStore database.

The API provides users the following benefits:

- Access to the MDCStore database.
- The ability to use the highest level of security and still be able to conduct business unrestricted.
- Access to the database error-handling mechanism.
- Access to all MDCStore privileges to retrieve data, manipulate data and save data to the database.
- Ease of use and robustness.
- C++ can be used to access the API.

Main Functionality

The MDCStoreUtils API provides a set of definitions, structures and enums and virtual pure callback classes. It also provides functions that work with the following:

- Database connections
- Results sets
- Results set data types
- Database errors
- BLOB data
- Measurement sets and results of measurement sets
- Plates
- Data acquisition
- Security features
- Datasets
- Images in measurement sets
- Quicklists
- Common database functions
- Application data tables
- File data

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

How to Use the MDCStoreUtils API

Supported database servers

MDCStoreUtils API supports two common database servers:

- Microsoft SQL Server 2005 and Microsoft SQL Server Express 2005
- Oracle 9.2 and higher

Compiling and linking with MDCStoreUtils API

Normally, the MDCStoreUtils API distribution comes with precompiled DLLs and all libraries needed. However, users may want to recompile the source files of this API distribution.

To compile and link application with MDCStoreUtils API:

- At compile time, include the follow header file anywhere to use the functions of MDCStoreUtils API:

```
# include “MDCStoreUtilsAPI.h”

# include “MDCSStructures.h”

# include “MDCSDefinitions.h”

# include “MDCSEnums.h”

# include “AxStringCollection.h”
```

- At link time, users need other libraries to link with MDCStoreUtils:

```
AxFileClient.lib

AxMFCDBUtils.lib

AxStringCollection.lib

MDCStoreUtils.lib
```

- At runtime, users need the following DLLs to run the program:

```
AxFileClient.dll

AxMFCDBUtils.dll

AxStringCollection.dll

MDCStoreUtils.dll
```

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Definitions

Dataset

A database query that returns a subset of objects in the database.

Instance

A sequence number that is used to denote multiple similar measurements at the same location.

Measurement Set

All the data extracted from one plate of images by a MetaXpress® application module with specific settings. Running the same application module again with different settings produces a separate measurement set.

Plate

All the images acquired from one physical microplate.

Result Set

A set of rows of data as a result of executing a SQL query.

Series

The location of an image or data value within a series in the z-dimension or the time dimension.

Site

A region within a microplate well from which an image has been acquired. Images can be acquired from more than one site per well.

Functions Available in MDCStoreUtils API

MDCStoreUtils API consists of functions divided into 15 categories. In this section, we provide all the function names, descriptions of what each function does based on these categories, and the function signatures.

There are two special functions you should always call before calling any function in MDCStoreUtils API:

To initialize the MDCStore interface:

```
BOOL MDCStoreUtils_Init()
```

To detach the MDCStore interface before exiting the application:

```
void WINAPI MDCStoreUtils_Finished()
```

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Database Connection Functions

2

This chapter contains functions that you can use to work with connections to databases.

Table 2-1: Database Connection Functions

Function Name	Description
MDCS_CONNECTION_GetAttributes	To get information about connection attributes
MDCS_CONNECTION_GetDetails	To obtain login information, then return a structure that could be used in all other function calls
MDCS_CONNECTION_TestCredentials	To test user credentials to see if they can create a database connection
MDCS_CONNECTION_CreateConnectionString	To generate ODBC connection string
MDCS_CONNECTION_GetDBHandleFromString	To get the database handle from a connection string
MDCS_CONNECTION_GetDBHandle	To get database handle to establish a connection to a database
MDCS_CONNECTION_GetNewDBHandle	To create a new database handle from an existing one
MDCS_CONNECTION_DestroyDBHandle	To destroy a database handle
MDCS_CONNECTION_Disconnect	To disconnect from the database
MDCS_CONNECTION_Reconnect	To reconnect to the database using an existing connection
MDCS_CONNECTION_GetDetailsFromString	To populate an MDCS_ST_UserLogin structure from an ODBC connection string
MDCS_CONNECTION_BeginTransaction	To start a transaction
MDCS_CONNECTION_CommitTransaction	To commit a transaction
MDCS_CONNECTION_RollbackTransaction	To rollback a transaction
MDCS_CONNECTION_GetDatabaseType	To get the type of database
MDCS_CONNECTION_GetInfo	To get information about a connection: Server name, type, version, ODBC name, driver version, etc.
MDCS_CONNECTION_SetAsyncMode	To set a database connection to async mode, this affects all statements
MDCS_CONNECTION_CancelQueryExecution(	To cancel the current execution statement
MDCS_CONNECTION_SetSilentMode	To set silent mode on a connection, no messages will be displayed
MDCS_CONNECTION_CheckIfDead	To check if a connection is active

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_CONNECTION_GetAttributes

```
BOOL MDCS_CONNECTION_GetAttributes(
HDBHANDLE hHandle,
MDCS_ST_ConnectionAttr& stConnectionAttr
);
```

Purpose

To get information about connection attributes; DSN (Data Source) name, User name and database name.

Parameters

hHandle - database connection handle

Output

stConnectionAttr - structure that contains connection attributes.

Return

False - if error occurred

MDCS_CONNECTION_GetDetails

```
BOOL MDCS_CONNECTION_GetDetails(
MDCS_ST_UserLogin * pstLoginInfo,
BOOL* pbSavePasswordChecked = NULL,
HWND hWnd = NULL,
LPCSTR pszTitle = NULL,
const MDCS_ST_UserLogin* pstLoginInfoIn = NULL,
BOOL bDefaultCheckSavePassword = FALSE
BOOL bReloginMode = FALSE
BOOL bValidateVersion = TRUE
);
```

Purpose

To get connection details by calling a dialog to obtain login information and return a structure that could be used in all other function calls. If called for Oracle, then the structure contains the selected database name.

Parameters

pstLoginInfoIn - default settings
bCheckSavePassword - default settings for password
pszTitle - dialog title
hWnd - application window
bDefaultCheckSavePassword - to check password by default
bReloginMode - bring up connection dialog in relogin mode

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Output

pstLoginInfo - selected settings

pbSavePasswordChecked - if password save option is checked, if null the option will not appear in the dialog

Return

FALSE - if dialog is cancelled

MDCS_CONNECTION_TestCredentials

```
BOOL MDCS_CONNECTION_TestCredentials(
const MDCS_ST_UserLogin * pstLoginInfo,
LPSTR pDatabaseName,
int nSize
BOOL bConnectAsAppUser = TRUE
);
```

Purpose

To test if the user can create a database connection with the supplied credentials.

Parameters

pstLoginInfo - pointer to the structure that contains login information

pDataBaseName - pointer to a database name

nSize - size of pDatabaseName

bConectAsAppUser - flag indicates that the connection should be made using application user permissions

Output

pDatabaseName - name of the database the connection is made to

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_CONNECTION_CreateConnectionString

```
BOOL MDCS_CONNECTION_CreateConnectionString(
const MDCS_ST_UserLogin * pstLoginInfo,
LPSTR pszConnectionString,
int nSize
);
```

Purpose

To generate an ODBC connection string from MDCS_ST_UserLogin structure.

Parameters

PstLoginInfo - pointer to the structure that contains login information
nSize - size of the connection string

Output

pszConnectionString - generated connection string

Return

FALSE - if error occurred

MDCS_CONNECTION_GetDBHandleFromString

```
HDBHANDLE MDCS_CONNECTION_GetDBHandleFromString(
LPCTSTER pszConnectionString
BOOL SetSilentMode = FALSE
);
```

Purpose

Function to generate a database connection handle from the ODBC connection string.

Parameters

SetSilentMode - flag that indicates if silent mode should be set on connection

pszConnectionString - ODBC connection string

Return

Database connection handle

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_CONNECTION_GetDBHandle

```
HDBHANDLE MDCS_CONNECTION_GetDBHandle(
const MDCS_ST_UserLogin& stLoginInfo
BOOL bSetSilentMode = FALSE
BOOL dConnectAsAppUser = TRUE
);
```

Purpose

Function to get a database handle.

Parameters

stLoginInfo - pointer to the structure that contains login information

bSetSilentMode - flag indicating if silent mode should be set on connection

dConnectAsAppUser - flag indicating that connection should be made using application user permissions

Return

Connection handle

MDCS_CONNECTION_GetNewDBHandle

```
HDBHANDLE MDCS_CONNECTION_GetNewDBHandle(
HDBHANDLE hDBHandle
BOOL bSetSilentMode = FALSE
);
```

Purpose

Function to create a new database handle from an existing one.

Parameters

hDBHandle - existing database connection handle

bSetSilentMode - flag indicating if silent mode should be set on connection

Return

Database connection handle

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_CONNECTION_DestroyDBHandle

```
void MDCS_CONNECTION_DestroyDBHandle(
HDBHANDLE hHandle
);
```

Purpose

Function to destroy the database handle.

Parameters

hHandle - database connection handle

MDCS_CONNECTION_Disconnect

```
void MDCS_CONNECTION_Disconnect(
HDBHANDLE hHandle
);
```

Purpose

Function to disconnect from the database.

Parameters

hHandle - database connection handle

MDCS_CONNECTION_Reconnect

```
BOOL MDCS_CONNECTION_Reconnect(
HDBHANDLE hHandle
);
```

Purpose

Function to reconnect to the database using the existing connection; can be used to make sure that the object is connected to the database.

Parameter

hHandle - database connection handle

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_CONNECTION_GetDetailsFromString

```
BOOL MDCS_CONNECTION_GetDetailsFromString(
MDCS_ST_UserLogin * pstLoginInfo,
LPCSTR pszConnectionString
);
```

Purpose

To generate the MDCS_ST_UserLogin structure from the ODBC connection string.

Parameters

pszConnectionString - ODBC connection string

Output

pstLoginInfo - pointer to the structure that contains login information

Return

FALSE - if error occurred

MDCS_CONNECTION_BeginTransaction

```
BOOL MDCS_CONNECTION_BeginTransaction(
HDBHANDLE hHandle
);
```

Purpose

To start a transaction.

Parameters

hHandle - database connection handle

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_CONNECTION_CommitTransaction

```
BOOL MDCS_CONNECTION_CommitTransaction(
HDBHANDLE hHandle
);
```

Purpose

To commit a transaction.

Parameters

hHandle - database connection handle

Return

FALSE - if error occurred

MDCS_CONNECTION_RollbackTransaction

```
BOOL MDCS_CONNECTION_RollbackTransaction(
HDBHANDLE hHandle
);
```

Purpose

To rollback a transaction.

Parameters

hHandle - database connection handle

Return

FALSE - if error occurred

MDCS_CONNECTION_GetDatabaseType

```
MDCS_E_SQLServerType MDCS_CONNECTION_GetDatabaseType(
HDBHANDLE hHandle
);
```

Purpose

To get the type of the database.

Parameters

hHandle - database connection handle

Return

MDCS_E_SQLServerType - type of database server (if the server type is supported)

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_CONNECTION_GetInfo

```
BOOL MDCS_CONNECTION_GetInfo(
HDBHANDLE hHandle,
MDCS_ST_ConnectionInfo& stConnectionInfo
);
```

Purpose

To get information about a connection such as: Server name, type, version, ODBC name, and driver version.

Parameters

hHandle - database connection handle

Output

stConnectionInfo - structure that contains connection information

Return

False - if error occurred

MDCS_CONNECTION_SetAsyncMode

```
BOOL MDCS_CONNECTION_SetAsyncMode(
      HDBHANDLE hHandle,
BOOL bSet = TRUE);
```

Purpose

To set the database connection to async mode. This affects all statements.

Parameters

hHandle - database connection handle

Output

None

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_CONNECTION_CancelQueryExecution(

```
BOOL MDCS_CONNECTION_CancelQueryExecution(
      HDBHANDLE hHandle
);
```

Purpose

To cancel the current execution statement.

Parameters

hHandle - database connection handle

Output

None

Return

FALSE - if error occurred

MDCS_CONNECTION_SetSilentMode

```
BOOL MDCS_CONNECTION_SetSilentMode(
      HDBHANDLE hHandle,
BOOL bSilentMode = TRUE
);
```

Purpose

To set silent mode on the connection – no messages will be displayed.

Parameters

hHandle - database connection handle

Output

None

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_CONNECTION_CheckIfDead

```
BOOL MDCS_CONNECTION_CheckIfDead(
      HDBHANDLE hHandle,
BOOL bReconnect = TRUE
);
```

Purpose

To check if the connection is dead.

Parameters

hHandle - database connection handle
bReconnect - reconnect if the connection is dead

Output

None

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Results Set Functions

3

This chapter contains descriptions of functions you can use to work with Results sets.

Table 3-1: Results Set Functions

Function Name	Description
MDCS_ImportDataToDB	To import result set data into the database
MDCS_GetResultsetData	To get result set data
MDCS_GetAllResultsets	To get a description of all result sets in the database
MDCS_GetResultsetInfo	To get a description of an individual result set

MDCS_ImportDataToDB

```
BOOL  MDCS_ImportDataToDB(
HDBHANDLE hHandle,
const MDCS_ST_ResultSet * pstObjDesc,
const MDCS_ImportResultSet * pExDataSource,
LPSTR pszObjectID,
int nSize,
MDCS_ProgressCallback * pCallBack
);
```

Purpose

To import the result set data into the database.

Parameters

pstObjDesc - result set description
pExDataSource - pointer to the data source that will supply data
hHandle- database connection handle
pCallBack - progress callback object
nSize - size of pszObjectID

Output

pszObjectID - ID of the newly created result set

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_GetResultsetData

```
BOOL MDCS_GetResultSetData (
HDBHANDLE hHandle,
LONGLONG lAssayID,
MDCS_GetDBResultsCCallback * pResultCallback
);
```

Purpose

To get result set data.

Parameters

hHandle - database connection handle
lAssayID - Assay ID that data will be returned for
pResultCallback - pointer to a callback function to get data,

Output

Includes the columns
PRINTED_SPOT_ID - feature ID
SUBSTANCE_NAME - feature name

Return

FALSE - if error occurred

Note: Call MDCS_GetResultsetDataTypes on page 37 to find out about the columns that are returned in a callback

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_GetAllResultsets

```
BOOL MDCS_GetAllResult sets(
HDBHANDLE hHandle,
MDCS_GetDBResultsCCallback * pResultCallback
);
```

Purpose

To get a description of all result sets in the database.

Parameters

hHandle - database connection handle
pResultCallback - pointer to a callback function to get data

Output

Columns from the ASSAYS table

Return

FALSE - if error occurred

MDCS_GetResultsetInfo

```
BOOL MDCS_GetResultsetInfo(
HDBHANDLE hHandle,
LONGLONG lAssayID,
MDCS_GetDBResultsCCallback * pResultCallback
);
```

Purpose

To get a description of individual result set.

Parameters

hHandle - database connection handle
lAssayID - Assay ID
pResultCallback - pointer to a callback function to get data

Output

Columns from the ASSAYS table

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Result Set Data Types Functions

4

This chapter contains functions you can use to work with result set data types.

Table 4-1: Result Set Data Type Functions

Function Name	Description
MDCS_GetResultsetDataTypes	To get result set data types
MDCS_GetAllDataTypes	To get all data types for all result sets.
MDCS_GetUniqueDataTypes	To get unique data types.
MDCS_GetAllDataTypesOfAssays	To get all data types of a list of assays.
MDCS_GetDataTypesNotInAssaysByDBName	To get all data types that do not belong to the set of assays.

MDCS_GetResultsetDataTypes

```
BOOL MDCS_GetResultsetDataTypes (
HDBHANDLE hHandle,
LONGLONG lAssayID,
MDCS_GetDBResultsCCallback * pResultCallback
);
```

Purpose

To get result set data types.

Parameters

hHandle - database connection handle

AssayID - Assay ID

pResultCallback - pointer to a call back function to get information from database

Output

Columns from the TABLE_COLUMNS table

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_GetAllDataTypes

```
BOOL MDCS_GetAllDataTypes(
HDBHANDLE hHandle,
MDCS_GetDBResultsCCallback * pResultCallback
);
```

Purpose

To get all data types for all result sets.

Parameters

hHandle - database connection handle

pResultCallback - pointer to a call back function to get information from database

Output

Columns from the TABLE_COLUMNS table

Return

FALSE - if error occurred

MDCS_GetUniqueDataTypes

```
BOOL MDCS_GetUniqueDataTypes(
HDBHANDLE hHandle,
MDCS_GetDBResultsCCallback * pResultCallback);
```

Purpose

To get unique data types.

Parameters

hHandle - database connection handle

pResultCallback - pointer to a call back function to get information from database

Output

Columns from the TABLE_COLUMNS table

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_GetAllDataTypesOfAssays

```
BOOL MDCS_GetAllDataTypesOfAssays (
HDBHANDLE hHandle,
LONGLONG arrAssayIDs,
      INT_PTR   nSize,
MDCS_GetDBResultsCCallback * pResultCallback
);
```

Purpose

To get all data types for all assays

Parameters

hHandle - database connection handle
arrAssayIDs - array of assays IDs
nSize - size of array (arrAssayIDs)
pResultCallback - pointer to a callback function to get data

Output

Columns from the TABLE_COLUMNS table

Return

FALSE - if error occurred

MDCS_GetDataTypesNotInAssaysByDBName

```
BOOL MDCS_GetAllDataTypesOfAssays (
HDBHANDLE hHandle,
LPCSTR pszDBColName,
LONGLONG* arrMicIDs,
INT_PTR nSize,
const MDCS_E_ColumnType& eColType,
const MDCS_E_AverageType& eAvgType,
MDCS_GetDBResultsCCallback* pResultCallback
);
```

Purpose

To get all data types that don’t belong to the provided measurement sets.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle
pszDBColName - DB name of data type
arrMicIDs- array of measurement set IDs
nSize - size of array arrMicIDs
eColType - format type of data type such as String or Float…
eAvgType - an average type of data type such as mean, minimal…
pResultCallback - pointer to a callback function to get data

Output

Columns from the TABLE_COLUMNS table
COLUMN_NAME, COLUMN_NAME_EXT, COLUMN_TYPE, AVG_TYPE, COLUMN_ID...

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Error Handling Functions 5

This chapter contains functions for working with error handling.

Table 5-1: Error Handling Functions

Function Name	Description
MDCS_GetLastError	To get last error and return its error code.
MDCS_GetLastErrorMsg	To get last error message and return error message
MDCS_GetDefaultErrorMsg	To get default error message for the error code

MDCS_GetLastError

```
int MDCS_GetLastError();
```

Purpose

To get the last error code from database.

Return

An error code (for example, MDCS_ERR_SUCCESS).

MDCS_GetLastErrorMsg

```
int MDCS_GetLastErrorMsg(
LPSTR pszError,
int nErrorSize
int* pnErrorSize = NULL
);
```

Purpose

To get the last error message.

Parameters

pszError - pointer to an error
nErrorSize - error size, could be as big as MDCS_MAX_ERR_SIZE
pnErrorSize - size of the error not including the null terminated symbol

Output

pszError - last error message

Return

An error code (for example, MDCS_ERR_SUCCESS).

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_GetDefaultErrorMsg

```
BOOL  MDCS_GetDefaultErrorMsg(
int nError,
LPSTR pszError,
int nSize
);
```

Purpose

Get default error message for the error code MDCS_ERR_XXXXXX.

Parameters

nError - error code
nSize - size of the pszError (can use MDCS_MAX_ERR_SIZE)

Output

pszError - error message

Return

FALSE - if error is not found

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

BLOB Functions

6

This chapter contains functions that you can use to work with Binary Large Objects (BLOBs).

Table 6-1: BLOB Functions

Function Name	Description
MDCS_BLOB_GetInfo	To get image information
MDCS_BLOB_GetLocationID	To get/create location ID
MDCS_BLOB_GetLocation	To get BLOB location description
MDCS_BLOB_Get	To get BLOB from the storage (database or file server)
MDCS_BLOB_Save	To save BLOB data in database or file server
MDCS_BLOB_RemoveBlobData	To remove BLOB data from database server and file server
MDCS_GetNewDatabaseID	To generate a new database ID
MDCS_BLOB_UpdateBlobDescAndName	To update BLOB’s description and name in database and file server
MDCS_BLOB_UpdateBlobDescAndNameEx	To update BLOB’s description and name (i.e. attached file) in database
MDCS_BLOB_SaveBlobEx	To save BLOB into database (i.e. attached file to object)
MDCS_BLOB_GetBLOBInfoEx	To get BLOB info using BLOB ID
MDCS_BLOB_GetInfoByReferenceID	To get BLOB info (i.e. image) using reference id
MDCS_BLOB_GetBlobInfoByReferenceIDEx	To get BLOB info (i.e. attached file) using reference id
MDCS_BLOB_GetNumBlobOfObject	To get number of BLOB of an object (i.e. dataset, measurement sets)
MDCS_BLOB_UpdatePlateObjectImageID	To update the image ID in the Plate image table using the BLOB ID
MDCS_BLOB_GetUniqueLocationNamebyFilter	To get a unique location name based on a filter
MDCS_BLOB_GetAllBlobInfoByLocationFilterAndLocationInfo	To get all image information based on criteria, BLOB type and file location filter
MDCS_BLOB_GetAllBlobInfoByPlateID	To get all image information based on the plate ID
MDCS_BLOB_DeleteUnUsedLocations	To remove unused locations in a File_location table
MDCS_BLOB_DeleteFileLocation	To remove a file location
MDCS_BLOB_LocationIsUsed	To check if any existing BLOB are stored in a location

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Table 6-1: BLOB Functions (cont'd)

MDCS_BLOB_RemoveBlobData	To remove BLOB data from datebase server and file server
MDCS_BLOB_RemoveFile	To remove file on file server or UNC path location
MDCS_BLOB_Attach	To attach a BLOB
MDCS_BLOB_UpdateBlobInfo	To update the description and display name of the attached file
MDCS_BLOB_MapLocations	To map BLOB locations
MDCS_BLOB_CanWriteToLocation	To check if the BLOB can be saved to a specified location
MDCS_BLOB_FillOutBlobLocationStruct	To fill out MDCS_ST_BlobLocation structure with results of MDCS_QueryResults

MDCS_BLOB_GetInfo

```
BOOL  MDCS_BLOB_GetInfo(
HDBHANDLE hHandle,
MDCS_ST_BlobInfo * pBlobInfo,
LONGLONG lBlobID,
MDCS_E_BlobType eBlobType
);
```

Purpose

To get image information using BLOB ID.

Parameters

hHandle - database connection handle

BlobID - database ID of the BLOB

eBlobType - BLOB Type

Output

pBlobInfo - BLOB information

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_BLOB_GetLocationID

```
BOOL  MDCS_BLOB_GetLocationID (
HDBHANDLE hHandle,
const MDCS_ST_BlobLocation * pstLocation,
LONGLONG* lLocationID,
BOOL bCreateNew = FALSE
);
```

Purpose

To get/create a location ID.

Parameters

hHandle - database connection handle
bCreateNew - if TRUE - function creates a new record
pstLocation - pointer to the structure that contains location description (all fields must be filled except for nLocationID)

Output

nLocationID - location ID, –1 if location is not found

Return

FALSE - if function fails

MDCS_BLOB_GetLocation

```
BOOL  MDCS_BLOB_GetLocation (
HDBHANDLE hHandle,
LONGLONG nLocationID,
MDCS_ST_BlobLocation * pstLocation
);
```

Purpose

To get the BLOB location description.

Parameters

hHandle - database connection handle
nLocationID - location ID

Output

pstLocation - pointer to the structure that contains location description

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_BLOB_Get

```
BOOL MDCS_BLOB_Get(
HDBHANDLE hHandle,
LONGLONG lBlobID,
MDCS_E_BlobType eBlobType,
MDCS_GetBlobCallback * pCallBack
);
```

Purpose

To get BLOB from storage, database or file server.

Parameters

hHandle - database connection handle
lBlobID - database ID of the BLOB
eBlobType - BLOB Type (from MDCS_E_BlobType)
pCallBack - pointer to callback used to transfer data

Return

FALSE - if error occurred

MDCS_BLOB_Save

```
BOOL  MDCS_BLOB_Save(
HDBHANDLE hHandle,
const MDCS_ST_BlobInfo * pImageInfo,
MDCS_E_BlobType eBlobType,
MDCS_SaveBlobCallback * pCallBack,
LONGLONG* lBlobIDOut,
const LONGLONG* lRefObjectID,
MDCS_E_FileStorage eStorage = MDCS_eDatabaseServer,
LPCSTR pszDirectoryExtra = NULL
);
```

Purpose

To save BLOB data in the database or file server.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle
pBlobInfo - pointer to the structure that contains image information
pCallBack - pointer to callback used to transfer data
eStorage - type of destination where image will be saved (default is database)
eBlobType - type of the BLOB
pszDirectoryExtra - if specified, creates a directory with pszDirectoryExtra on FileServer

Output

lBlobIDOut - database ID of the BLOB
lRefObjectID - object reference ID

Return

FALSE - if error occurred

MDCS_BLOB_RemoveBlobData

```
BOOL  MDCS_BLOB_RemoveBlobData(
HDBHANDLE hHandle,
LONGLONG lID,
MDCS_E_BlobType eBlobType
);
```

Purpose

To remove BLOB data from database server and file server.

Parameters

hHandle - database connection handle
lID - ID of the record that needs to be deleted
eBlobType - type of the BLOB

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_GetNewDatabaseID

```
BOOL  MDCS_GetNewDatabaseID(
HDBHANDLE hHandle,
LPSTR pszID,
int nIDSize,
int* pnIDSizeOut
);
```

Purpose

To generate a new database ID.

Parameters

hHandle - database connection handle
nIDSize - size of ID (use MDCS_MAX_DATABASE_ID)

Output

pszID - generated ID
pnIDSizeOut - size of the generated ID

Return

FALSE - if error occurred

MDCS_BLOB_UpdateBlobDescAndName

```
BOOL  MDCS_BLOB_UpdateBlobDescAndName(
HDBHANDLE hHandle,
LONGLONG lBlobID,
Const MDCS_ST_BlobInfo* pBlobInfo,
MDCS_E_BlobType eBlobType
);
```

Purpose

To update a BLOB's description and name.

Parameters

hHandle - database connection handle
lBlobID - Blob ID in database
pBlobInfo – Blob info to update in the database (only need description and name fields)
eBlobType – Blob type

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_BLOB_UpdateBlobDescAndNameEx

```
BOOL  MDCS_BLOB_UpdateBlobDescAndName(
HDBHANDLE hHandle,
LONGLONG lBlobID,
Const MDCS_ST_BlobInfoEx* pBlobInfo,
MDCS_E_BlobType eBlobType
);
```

Purpose

Function to update a BLOB's description and name.

Parameters

hHandle - database connection handle
lBlobID - Blob ID in database
pBlobInfo – Blob info to update in database (only need description and name fields)
eBlobType – Blob type

Return

FALSE - if error occurred

MDCS_BLOB_SaveBlobEx

```
BOOL  MDCS_BLOB_SaveBlobEx(
HDBHANDLE hHandle,
Const MDCS_ST_BlobInfoEx* pBlobInfo,
MDCS_E_BlobType eBlobType,
MDCS_SaveBlobCallback* pCallBack,
LONGLONG* lBlobIDOut,
Const LONGLONG* lRefObjectID
);
```

Purpose

To save a BLOB into the database or file server.

Parameters

hHandle - database connection handle
pBlobInfo – Blob info (ID not required, since this is input)
eBlobType – Blob type
pCallBack - callback class to save BLOB
lRefObjectID – reference object id that the BLOB associated with

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

OUTPUT

lBlobIDOut – output id of saved BLOB

Return

FALSE - if error occurred

MDCS_BLOB_GetBLOBInfoEx

```
BOOL MDCS_BLOB_GetBLOBInfoEx(
HDBHANDLE hHandle,
MDCS_ST_BlobInfoEx* pBlobInfo,
LONGLONG* lBlobID,
MDCS_E_BlobType eBlobType
);
```

Purpose

To get attached file or BLOB information based on provided BLOB ID.

Parameters

hHandle - database connection handle
eBlobType – BLOB type
lBlobID - BLOB ID

OUTPUT

pBlobInfo – BLOB info

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_BLOB_GetInfoByReferenceID

```
BOOL  MDCS_BLOB_ GetInfoByReferenceID (
HDBHANDLE hHandle,
LONGLONG lReferenceID,
MDCS_E_BlobType eBlobType,
MDCS_GetDBResultsCCallback* pResultCallback
);
```

Purpose

To get BLOB information based on provided reference id (dataset ID/ assay ID).

Parameters

hHandle - database connection handle
lReferenceID – reference id such as dataset id or measurement set id
eBlobType – BLOB type
pResultCallback - pointer to a callback function to get data

OUTPUT

Columns for BLOB Info like ID, SIZE, DESC etc.

Return

FALSE - if error occurred

MDCS_BLOB_GetBlobInfoByReferenceIDEx

```
BOOL  MDCS_BLOB_ GetInfoByReferenceIDEx(
HDBHANDLE hHandle,
LONGLONG lReferenceID,
MDCS_E_BlobType eBlobType,
MDCS_GetDBResultsCCallback* pResultCallback
);
```

Purpose

Function to get BLOB information from the database using BLOB id and type.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle
lReferenceID – reference id such as dataset id or measurement set id
eBlobType – BLOB type
pResultCallback - pointer to a callback function to get data

Output

Columns for BLOB info like ID, SIZE, DESC etc.

Return

FALSE - if error occurred

MDCS_BLOB_GetNumBlobOfObject

```
BOOL  MDCS_BLOB_GetNumBlobOfObject(
HDBHANDLE hHandle,
LONGLONG lReferenceID,
MDCS_E_BlobType eBlobType,
LONGLONG& lNumbBlob
);
```

Purpose

Function to get the number of BLOB files attached to an object (assay, dataset) using the reference ID.

Parameters

hHandle - database connection handle
lReferenceID – reference id such as dataset id or assay id
eBlobType – BLOB type

OUTPUT

lNumbBlob – number of BLOB files associated with the object (dataset, measurement set)

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_BLOB_UpdatePlateObjectImageID

```
BOOL  MDCS_BLOB_UpdatePlateObjectImageID(
HDBHANDLE hHandle,
LONGLONG lCurrentImageObjID,
LONGLONG lNewImageObjID,
);
```

Purpose

To update the image ID in the PLATE_IMAGES table using the BLOB ID.

Parameters

hHandle - database connection handle
lCurrentImageObjID - BLOB ID
lNewImageObjID -Updated value of BLOB ID in PLATE_IMAGES

Return

FALSE - if error occurred

MDCS_BLOB_GetUniqueLocationNamebyFilter

```
BOOL  MDCS_BLOB_GetUniqueLocationNamebyFilter (
HDBHANDLE hHandle,
MDCS_CL_BlobLocationCB* pCallBack
MDCS_E_LocationFilter eFilter = MDCS_eServerName
);
```

Purpose

To get unique location information based on a filter.

Parameters

eFilter - filter to get data
hHandle - database connection handle
pCallBack - callback object

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Output

Depending on the filter, the output result set will contain one, some, or all (1-5) of the following:

1) server name

2) serverRoot

3) Directory

4) Location type

5) port number

Return

FALSE - if function fails

MDCS_BLOB_GetAllBlobInfoByLocationFilterAndLocationInfo

```
BOOL
MDCS_BLOB_GetAllBlobInfoByLocationFilterAndLocationInfo (
HDBHANDLE hHandle,
MDCS_ST_BlobLocation* arrBlobLocation,
INT_PRT NNumLocation,
MDCS_E_BlobType BlobType
MDCS_CL_BlobInfoCB* PCallBack
MDCS_E_LocationFilter eFilter = MDCS_eServerName
);
```

Purpose

To get all image information based on criteria, BLOB type and file location filter (server name, root name, or directory).

Parameters

eFilter - filter to get data

hHandle - database connection handle

eBlobType - Type of BLOB

arrBlobLocation - all BLOB Location information

NNumLocation - size of arrBlobLocation array

Output

pCallback - call back result

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_BLOB_GetAllBlobInfoByPlateID

```
BOOL  MDCS_BLOB_GetAllBlobInfoByPlateID (
HDBHANDLE hHandle,
LONGLONG lPlateID,
MDCS_E_BlobType eBlobType
MDCS_CL_BlobInfoCB* PCallBack
);
```

Purpose

To get all image information based on the plate ID.

Parameters

hHandle - database connection handle
lPlateID - plate ID
eBlobType - Type of BLOB

Output

pCallback - call back result

Return

FALSE - if error occurred

MDCS_BLOB_DeleteUnUsedLocations

```
BOOL  MDCS_BLOB_DeleteUnUsedLocations(
HDBHANDLE hHandle,
MDCS_ProgressCallBack* pProgressCallback
);
```

Purpose

To remove unused locations in FILE_LOCATIONS.

Parameters

hHandle - database connection handle
pProgressCallback - progress callback

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_BLOB_DeleteFileLocation

```
BOOL  MDCS_BLOB_DeleteFileLocation(
HDBHANDLE hHandle,
const LONGLONG lLocationID
);
```

Purpose

To remove a file location.

Parameters

hHandle - database connection handle
lLocationID - Location ID

Return

FALSE - if error occurred

MDCS_BLOB_LocationIsUsed

```
BOOL  MDCS_BLOB_LocationIsUsed (
HDBHANDLE hHandle,
LONGLONG lLocationID,
BOOL& bIsUsed
);
```

Purpose

To check if any existing BLOBs are stored in the location.

Parameters

hHandle - database connection handle
lLocationID - location ID

Output

bIsUsed - TRUE if the Location ID is used by a BLOB, otherwise FALSE

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_BLOB_RemoveBlobData

```
BOOL  MDCS_BLOB_RemoveBlobData (
HDBHANDLE hHandle,
LONGLONG lID,
MDCS_E_BlobType eBlobType
BOOL bNotRemoveRecord = FALSE
);
```

Purpose

To remove BLOB data from a database, file server or UNC path location.

Parameters

hHandle - database connection handle
lID - ID of the record that needs to be deleted
eBlobType - Type of BLOB
bNotRemoveRecord - Whether to delete the image record or not

Return

FALSE - if error occurred

MDCS_BLOB_RemoveFile

```
BOOL  MDCS_BLOB_RemoveFile (
HDBHANDLE hHandle,
const MDCS_ST_BlobLocation& stBlobLocation
LPCSTR szFileName
);
```

Purpose

To remove a file from the image storage location (database, file server or UNC path).

Parameters

hHandle - database connection handle
stBlobLocation - structure describing the location of the file to be deleted
eBlobType - Type of BLOB

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_BLOB_Attach

```
BOOL  MDCS_BLOB_Attach (
MDCS_E_BlobType eBlobType
const MDCS_ST_BlobinfoEX& stBlobInfo,
LONGLONG* plIDOut = NULL
);
```

Purpose

To attach a BLOB (change BLOB information).

Parameters

hHandle - database connection handle
stBlobInfo - structure that describes BLOB.
eBlobType - Type of BLOB

Output

plIDOut - created image ID

Return

FALSE - if error occurred

MDCS_BLOB_UpdateBlobInfo

```
BOOL  MDCS_BLOB_UpdateBlobInfo (
HDBHANDLE hHandle,
LONGLONG lBlobID
const MDCS_ST_BlobInfo* pBlobInfo,
MDCS_E_BlobType eBlobType
);
```

Purpose

To update the description and display name of the BLOB.

Parameters

hHandle - database connection handle
lBlobID - BLOB ID
pBlobInfo - information to update in database
eBlobType - Type of BLOB

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_BLOB_MapLocations

```
BOOL  MDCS_BLOB_MapLocations (
HDBHANDLE hHandle,
const MDCS_E_BlobLocation& stSource
const MDCS_ST_BlobLocation& stDest
);
```

Purpose

To map BLOB locations.

Parameters

hHandle - database connection handle
stSource - source BLOB location
stDest - destination BLOB location

Return

FALSE - if error occurred

MDCS_BLOB_CanWriteToLocation

```
BOOL  MDCS_BLOB_CanWriteToLocation(
HDBHANDLE hHandle,
const MDCS_STBlobLocation& stBlobLocation
LPCSTR pszDirectoryExtra = NULL
);
```

Purpose

To check if the BLOB can be saved to a specified location.

Parameters

hHandle - database connection handle
stBlobLocation - structure that describes the location
pszDirectoryExtra - if specified, will create a directory with pszDirectoryExtra on File server

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_BLOB_FillOutBlobLocationStruct

```
BOOL  MDCS_BLOB_FillOutBlobLocationStruct(
const MDCS_QueryResults* pResults,
MDCS_ST_BlobLocation& stBlobLoc,
BOOL bvNativeFormat = TRUE
);
```

Purpose

To fill out MDCS_ST_BlobLocation structure with results of MDCS_QueryResults.

Parameters

pResults - results of the query

bvNativeFormat - format type of query results, if not native then all results are treated as strings

Output

stBlobLoc - location structure (if stBlobLoc.lLocationID == 0 - the BLOB is in the database)

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Measurement Sets and Results Functions

7

This chapter contains functions that you can use to work with measurement sets and results of measurement sets.

Table 7-1: Measurement Sets and Results of Measurement Sets Functions

Function Name	Description
MDCS_ASSAY_AppendMeasurementSet	To import (append) measurement set data into the database
MDCS_ASSAY_ImportMeasurementSet	To remove shape data from a measurement set
MDCS_ASSAY_ImportMeasurementSet	To import measurement set data into the database
MDCS_ASSAY_InsertDValue	To insert value of double type to the table that contains results of measurement set
MDCS_ASSAY_InsertNumericValue	To insert value of double type to the table that contains results of measurement set
MDCS_ASSAY_InsertSValue	To insert value of string to the table that contains results of measurement set
MDCS_ASSAY_InsertStringValue	To insert value of string type to the table that contains results of measurement set
MDCS_ASSAY_UpdateShapeLines	To insert or update shapes
MDCS_ASSAY_GetShapeLineBlobDesc	To get shape lines blob description
MDCS_ASSAY_GetShapeLineBlobDescBySite	To get shape lines blob description per site
MDCS_ASSAY_GetShapeLineBlobDescBySiteAndSeries	To get shape lines blob description per site and series
MDCS_ASSAY_GetShapeLineBlobDescBySiteSeriesAndInstance	To get shape lines blob description per site, series and instance
MDCS_ASSAY_GetShapeLineBlobDescByPlate	To get shape lines blob description by plate
MDCS_ASSAY_GetShapeLineBlobPerAssay	To get all the shape lines blob descriptions per measurement set
MDCS_ASSAY_GetShapeLines	To get shape lines of a shape
MDCS_ASSAY_GetMeasurementAttributes	To get a measurement description based on measurement set ID and measurement ID
MDCS_ASSAY_GetAssayMeasurementRecord	To get a measurement record

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Table 7-1: Measurement Sets and Results of Measurement Sets Functions (cont’d)

MDCS_ASSAY_GetAssayMeasurementsByPlateAndMeasurement	To get measurement set measurement by plate ID and measurement ID
MDCS_ASSAY_GetMeasurementsBySiteID	To get measurements by Site ID
MDCS_ASSAY_GetMeasurementInfoByAssayAndColumnName	To get a measurement description based on measurement set ID and database column name
MDCS_ASSAY_GetMeasurementByFunctionAndParameterName	To get a measurement attribute description based on measurement and functions names
MDCS_ASSAY_GetMeasurementRecord	To get a data type record by its ID
MDCS_ASSAY_Create	To create a new measurement set in a specific folder
MDCS_ASSAY_GetAssaySiblingFolders	To get measurement set sibling folders
MDCS_ASSAY_CreateMeasurement	To create a new measurement set measurement
MDCS_ASSAY_AddMeasurement	To add a new measurement to a measurement set
MDCS_ASSAY_GetMeasurementByID	To get the description of a measurement set measurement
MDCS_ASSAY_GetMeasurementByName	To get the description of a measurement set measurement
MDCS_ASSAY_Delete	To delete a measurement set
MDCS_ASSAY_DeleteAllForPlate	To delete all measurement sets for a plate
MDCS_ASSAY_DeleteDataForPlate	To delete measurement set results for a plate
MDCS_ASSAY_DeleteFolder	To delete a measurement set folder
MDCS_ASSAY_CreateOutlinesTable	To create a measurement set outlines table
MDCS_ASSAY_ManageFolderSecurity	Call a dialog to manage security access to the measurement set folders
MDCS_ASSAY_CanModifyFolder	To check if a user can modify a folder
MDCS_ASSAY_CreateFolder	Create a new measurement set folder in the database
MDCS_ASSAY_ModifyFolder	Modify a measurement set folder in the database
MDCS_ASSAY_DoesSubFolderExist	To check if the sub folder exists

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Table 7-1: Measurement Sets and Results of Measurement Sets Functions (cont'd)

MDCS_ASSAY_GetAllInFolder	To get measurement sets with a basic description in folder
MDCS_ASSAY_GetFolderPath	To get the path of a folder
MDCS_ASSAY_Reindex	To reindex measurement set measurements
MDCS_ASSAY_ReindexShapes	To reindex measurement set cell shapes
MDCS_ASSAY_OptimizeAll	To optimize all measurement set tables
MDCS_ASSAY_Copy	To copy a measurement set
MDCS_ASSAY_Merge	To merge a measurement set
MDCS_ASSAY_GetSiteCount	To count site appearance in measurement set table
MDCS_ASSAY_GetSiteInfoImageByID	To get site image information by provided ID
MDCS_ASSAY_GetSiteInfoImageBySiteAndSeriesID	To get site image info by site and series ID
MDCS_ASSAY_MeasurementGetSiteCount	To count site appearance in a measurement set
MDCS_ASSAY_DeleteMeasurement	To delete a measurement
MDCS_ASSAY_CellOutlinesGetSiteCount	To count site appearance in a assay cell outlines
MDCS_ASSAY_CreateNewName	Create a new measurement set name in the destination folder.
MDCS_ASSAY_CreateRun	To create a new measurement set run record
MDCS_ASSAY_GetLatestAssayRunID	To get the ID of the latest run on a measurement set
MDCS_ASSAY_GetByID	Get measurement set data
MDCS_ASSAY_CreateProfile	Create a new measurement set profile record
MDCS_ASSAY_AssociateWithPlate	To associate a measurement set with a plate
MDCS_ASSAY_GetSpotID	To get the spot ID from instance site and series
MDCS_ASSAY_GetAllForPlate	To get all measurement sets that are associated with a plate
MDCS_ASSAY_GetProfiles	To get all available profiles
MDCS_ASSAY_GetProfile	To get a profile record
MDCS_ASSAY_GetProfileInfo	To get profile information
MDCS_ASSAY_UpdateProfile	To update a profile record
MDCS_ASSAY_DeleteProfile	To delete a measurement set profile

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Table 7-1: Measurement Sets and Results of Measurement Sets Functions (cont'd)

MDCS_ASSAY_GetRecord	To get a measurement set record
MDCS_ASSAY_GetAssayByRunID	To get a measurement set record using RUN ID
MDCS_ASSAY_UsedInDatasets	To check if a measurement set is used in datasets
MDCS_ASSAY_HavePermissionsToModify	To check if the current user can modify a measurement set
MDCS_ASSAY_GetImageSourceRecords	To get all records from the IMAGE_SOURCE table
MDCS_ASSAY_GetStatisticalValuesForPlateAndAssay	To get statistical values for a selected plate and assay
MDCS_ASSAY_GetStatisticalValuesForPlateAndMeasurement	To get statistical values for a selected plate and measurement
MDCS_ASSAY_GetValuesForPlateAndAssay	To get values for the selected plate and assay
MDCS_ASSAY_GetValuesForPlate	To get values for the selected plate
MDCS_ASSAY_GetAllSiteMeasurements	To get all values for the selected site
MDCS_ASSAY_GetMeasurementInfoByAssayAndPlate	To get data types records by plate and assay
MDCS_ASSAY_GetAssayInfoByNamePlateAndSettings	To get assay information by plate name and plate setting
MDCS_ASSAY_GetAssayIDsOfPlate	To get assay IDs that are available for a plate
MDCS_ASSAY_GetAssaySettingsOfPlate	To get unique assay settings of the plate
MDCS_ASSAY_GetMeasurementStatistic	To get statistic use of a measurement (number of assays and datasets that use a measurement set)
MDCS_ASSAY_GetShapeLinesBySite	To get shapeline blob description per site.
MDCS_ASSAY_GetShapeLinesBySiteAndSeries	To get shapeline blob description per site and series.
MDCS_ASSAY_GetMarkedAssaysWithCallback	To get assays marked for deletion
MDCS_ASSAY_Restore	To restore the deleted assay
MDCS_ASSAY_CreateForPlate	To create a new assay for a specific plate.
MDCS_ASSAY_GetMeasurementByName	To get the description of an assay measurement using display name
MDCS_ASSAY_GetMeasurementByDBName	To get the description of an assay measurement using DB column name

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Table 7-1: Measurement Sets and Results of Measurement Sets Functions (cont’d)

MDCS_ASSAY_GetScopeAttributeByName	To get the description of a scope attribute using attribute display name
MDCS_ASSAY_GetScopeAttributeByID	To get the description of a scope attribute using DB column name
MDCS_ASSAY_DeleteMeasurementInAssays	To delete a measurement from all assays that use it but not the default
MDCS_ASSAY_UpdateDataType	To update a measurement property
MDCS_ASSAY_UpdateDataTypeByAssay	To modify a measurement that is not a default measurement for an assay based on DB column name and assay id
MDCS_ASSAY_CanModifyAssay	To check if the current user can modify an assay
MDCS_ASSAY_CalculateStatisticResults	To get statistic results on an array of assay ids
MDCS_ASSAY_CalculateStatisticEx	To get statistic results on an array of assay ids
MDCS_ASSAY_GetZPrime	To calculate Z' on assay and measurement type
MDCS_ASSAY_GetZPrimeScopeAttribute	To calculate Z' on assay and scope attribute
MDCS_ASSAY_GetUniqueMeasurementValues	To get unique values of measurement type for array of mensuration sets
MDCS_ASSAY_FindMeasurementValues	To find values from a provided array in a measurement set
MDCS_ASSAY_UpdateMeasurementData	To update measurement data based on transformation criteria
MDCS_ASSAY_GetAllMSetUniqueAnnotation	To get unique annotation of all assays
MDCS_ASSAY_CreateAttribute	To create an Assay attribute
MDCS_ASSAY_GetAttributeInfoByDisplayName	To get attribute info using its display name.
MDCS_ASSAY_GetAttributeInfoByDBName	To get attribute information using its database name
MDCS_ASSAY_GetAttributeValueByDBName	To get the assay attribute value using its database column name
MDCS_ASSAY_GetAttributeValueByDisplayName	To get the assay attribute value using its display name
MDCS_ASSAY_AssignAttributeValueString	To assign an Assay attribute value (type string)
MDCS_ASSAY_AssignAttributeValueLong	To assign an Assay attribute value (type Long)

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Table 7-1: Measurement Sets and Results of Measurement Sets Functions (cont'd)

MDCS_ASSAY_AssignAttributeValueFloat	To assign an Assay attribute value (type Float)
MDCS_ASSAY_RenameAttribute	To rename an Assay attribute
MDCS_ASSAY_DeleteAttribute	To delete an Assay attribute based on its DB name
MDCS_ASSAY_GetHeaderAndFileInfo	To get header info and file import location of a measurement set
MDCS_ASSAY_UpdateMeasurementSetName	To update a measurement set name
MDCS_ASSAY_UpdateMeasurementSetDescription	To update a measurement set description
MDCS_ASSAY_GetUniqueMeasurementValuesByPlate	To get unique values of measurement types for a plate

MDCS_ASSAY_AppendMeasurementSet

```
BOOL  MDCS_ASSAY_AppendMeasurementSet(
HDBHANDLE hHandle,
const MDCS_CL_ImportDS * pDatasource,
MDCS_ProgressCallback* pCallback
);
```

Purpose

To import measurement set data into the database.

Parameters

hHandle - database handle
pDatasource - pointer to a datasource data
pCallBack - progress callback class

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_ImportMeasurementSet

```
BOOL  MDCS_ASSAY_RemoveShapeData(
HDBHANDLE hHandle,
LONGLONG lAssayID,
);
```

Purpose

To remove shape data from a measurement set.

Parameters

hHandle - database handle
lAssayID - assay ID

Return

FALSE - if error occurred

MDCS_ASSAY_ImportMeasurementSet

```
BOOL  MDCS_ASSAY_ImportMeasurementSet(
HDBHANDLE hHandle,
LONGLONG lAssayID,
MDCS_ImportMeasurementSet* pDatasource,
MDCS_ProgressCallback* pCallback
);
```

Purpose

To import measurement set data into the database.

Parameters

hHandle - database handle
lAssayID - assay ID
pDatasource - pointer to a datasource data
pCallBack - progress callback class

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_InsertDValue

```
BOOL  MDCS_ASSAY_InsertDValue(
HDBHANDLE hHandle,
const MDCS_ST_ShapeInfo* stShapeInfo,
LONGLONG lAssayID,
LONGLONG lMeasurementID,
double   dValue
);
```

Purpose

To insert the value of a double type to the table that contains results of a measurement set.

Parameters

hHandle - database connection handle
pstShapeInfo - pointer to a structure that describes shape
lAssayID - ID of an assay that was run
MeasurementID - measurement set ID of an assay
dValue - value to insert

Return

FALSE - if error occurred

MDCS_ASSAY_InsertNumericValue

```
BOOL  MDCS_ASSAY_InsertNumericValue(
HDBHANDLE hHandle,
LPCSTR pszTableName,
LPCSTR pszColumnName,
const MDCS_ST_ShapeInfo* stShapeInfo,
double dValue
);
```

Purpose

To insert the value of a double type to the table that contains results of a measurement set.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle
pstShapeInfo - pointer to a structure that describes shape
pszTableName - table name where values are inserted
pszColumnName - column name of a measurement
dValue - value to insert

Return

FALSE - if error occurred

MDCS_ASSAY_InsertSValue

```
BOOL  MDCS_ASSAY_InsertSValue(
DBHANDLE hHandle,
const MDCS_ST_ShapeInfo* pstShapeInfo,
LONGLONG lAssayID,
LONGLONG lMeasurementID,
LPCSTR pszValue
);
```

Purpose

To insert the value of a string to the table that contains results of a measurement set.

Parameters

hHandle - database connection handle
pstShapeInfo - pointer to a structure that describes shape
lAssayID - ID of an assay that was run
lMeasurementID - measurement set ID of an assay
pszValue - value to insert

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_InsertStringValue

```
BOOL  MDCS_ASSAY_InsertStringValue(
HDBHANDLE hHandle,
LPCSTR pszTableName,
LPCSTR pszColumnName,
const MDCS_ST_ShapeInfo* stShapeInfo,
LPCSTR pszValue
);
```

Purpose

To insert the value of a string type to the table that contains results of measurement set.

Parameters

hHandle - database connection handle
pstShapeInfo - pointer to a structure that describes shape
pszTableName - table name where values are inserted
pszColumnName - column name of a measurement
pszValue - value to insert

Return

FALSE - if error occurred

MDCS_ASSAY_InsertShapeLines

```
BOOL  MDCS_ASSAY_InsertShapeLines(
HDBHANDLE hHandle,
LPCSTR pszTableName,
const MDCS_ST_ShapeInfo* pstShapeInfo,
const MDCS_ST_ShapeLine* pShapeLines,
UINT nNumShapeLines
);
```

Purpose

To insert or update shapes.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle

pstShapeInfo - pointer to a structure that describes shape

pszTableName - table name where values are inserted

pShapeLines - pointer to the array of structures that describes shape lines

nNumShapeLines - number of the shape lines in the array pShapeLines

Return

FALSE - if error occurred

MDCS_ASSAY_UpdateShapeLines

```
BOOL  MDCS_ASSAY_UpdateShapeLines(
HDBHANDLE hHandle,
LPCSTR pszTableName,
LONGLONG lShapeID,
const MDCS_ST_ShapeInfo* pstShapeInfo,
const MDCS_ST_ShapeLine* pShapeLines,
UINT nNumShapeLines
);
```

Purpose

To update shape lines.

Parameters

hHandle - database connection handle

lShapeID - shape ID to update

pstShapeInfo - pointer to a structure that describes shape

pszTableName - table name where values are inserted

pShapeLines - pointer to the array of structures that describe shape lines

nNumShapeLines - number of the shape lines in the array (pShapelines)

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_GetShapeLineBlobDesc

```
BOOL  MDCS_ASSAY_GetShapeLineBlobDesc(
HDBHANDLE hHandle,
LPCSTR pszTableName,
const MDCS_ST_ShapeInfo* pstShapeInfo,
MDCS_ST_ShapeLinesBlob *pBlobDesc,
LONGLONG* plRecordsFound
);
```

Purpose

To get shape lines BLOB description.

Parameters

hHandle - database connection handle
pstShapeInfo - pointer to a structure that describes shape
pszTableName - table name where values are inserted
pBlobDesc - pointer to the structure that describes a blob of shape lines

Output

plRecordsFound - number of records found

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_GetShapeLineBlobDescBySite

```
BOOL  MDCS_ASSAY_GetShapeLineBlobDescBySite(
HDBHANDLE hHandle,
LPCSTR pszTableName,
LONGLONG lSiteID
MDCS_GetDBResultsCCallback* pResultCallback
);
```

Purpose

To get shape lines BLOB description per site.

Parameters

hHandle - database connection handle
pszTableName - table name where values are inserted
lSiteID - site ID to get BLOBs for
pResultCallback - pointer to callback function to get data

Output

Columns:
SITE_ID
SERIES_ID
INSTANCE
SHAPE_TYPE
COLOR
LINES_TOTAL - total lines counted for the shapes per combination (SITE_ID, SERIES_ID, INSTANCE, SHAPE_TYPE, COLOR)
RECORD_COUNT - total records found for the shapes per combination above
BLOB_SIZE - data size to retrieve for the shapes per combination above

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_GetShapeLineBlobDescBySiteAndSeries

```
BOOL  MDCS_ASSAY_GetShapeLineBlobDescBySiteAndSeries(
HDBHANDLE hHandle,
LPCSTR pszTableName,
LONGLONG lSiteID,
LONGLONG lSeriesID,
MDCS_GetDBResultsCCallback* pResultCallback
);
```

Purpose

To get shape lines BLOB description by site and series.

Parameters

hHandle - database connection handle
pszTableName - table name where values are inserted
lSiteID - site ID where to get BLOBs from
lSeriesID - series ID
pResultCallback - pointer to a callback function to get data

Output

Columns:

SITE_ID
SERIES_ID
INSTANCE
SHAPE_TYPE
COLOR
LINES_TOTAL - total lines counted for the shapes per combination (SITE_ID, SERIES_ID, INSTANCE, SHAPE_TYPE, and COLOR)

RECORD_COUNT - total records found for the shapes per combination above

BLOB_SIZE - data size to retrieve for the shapes per combination above

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_GetShapeLineBlobDescBySiteSeriesAndInstance

```
BOOL
MDCS_ASSAY_GetShapeLineBlobDescBySiteSeriesAndInstance(
HDBHANDLE hHandle,
LPCSTR pszTableName,
LONGLONG lSiteID,
LONGLONG lSeriesID,
LONGLONG lInstanceID,
MDCS_GetDBResultsCCallback* pResultCallback
);
```

Purpose

To get shape lines BLOB description per site, series and instance.

Parameters

hHandle - database connection handle
pszTableName - table name where values are inserted
lSiteID - site ID where to get BLOBs from
lSeriesID - series ID
lInstance - instance ID
pResultCallback - pointer to callback function to get data

Output

Columns*:*
SITE_ID - site ID
SERIES_ID
INSTANCE
SHAPE_TYPE
COLOR
LINES_TOTAL - total lines counted for the shapes per combination (SITE_ID, SERIES_ID, INSTANCE, SHAPE_TYPE, and COLOR)
RECORD_COUNT - total records found for the shapes per combination above
BLOB_SIZE - data size to retrieve for the shapes per combination above

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_GetShapeLineBlobDescByPlate

```
BOOL  MDCS_ASSAY_GetShapeLineBlobDescByPlate(
HDBHANDLE hHandle,
LPCSTR pszTableName,
LONGLONG lPlate,
MDCS_GetDBResultsCCallback* pResultCallback
);
```

Purpose

To get shape lines BLOB description per plate.

Parameters

hHandle - database connection handle
pszTableName - table name where values are inserted
IPlateID - plate ID
pResultCallback - pointer to callback function to get data

Output

Columns:
SITE_ID - site ID
SERIES_ID
INSTANCE
SHAPE_TYPE
COLOR
LINES_TOTAL - total lines counted for the shapes per combination SITE_ID, SERIES_ID, INSTANCE, SHAPE_TYPE, and COLOR
RECORD_COUNT - total records found for the shapes per combination above
BLOB_SIZE - data size to retrieve for the shapes per combination above

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_GetShapeLineBlobPerAssay

```
BOOL  MDCS_ASSAY_GetShapeLineBlobPerAssay(
HDBHANDLE hHandle,
LPCSTR pszShapeName,
MDCS_GetDBResultsCCallback* pResultCallback
);
```

Purpose

To get all the shape descriptions per assay.

Parameters

hHandle - database connection handle
pszShapeName - table name where shapes are stored
pResultCallback - pointer to callback function to get data

Output

Columns:
SITE_ID - site ID
SERIES_ID
INSTANCE
SHAPE_TYPE
COLOR
LINES_TOTAL - total lines counted for the shapes per combination SITE_ID, SERIES_ID, INSTANCE, and SHAPE_TYPE
RECORD_COUNT - total records found for the shapes per combination above
BLOB_SIZE - data size to retrieve for the shapes per combination above

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_GetShapeLines

```
BOOL  MDCS_ASSAY_GetShapeLines(
HDBHANDLE hHandle,
LPCSTR pszTableName,
const MDCS_ST_ShapeInfo* pstShapeInfo,
MDCS_ST_ShapeLine *pShapeLines,
LONGLONG lNumShapeLines
);
```

Purpose

To get shape lines of a shape.

Parameters

hHandle - database connection handle
pstShapeInfo - pointer to a structure that describes shape
pszTableName - table name where values are inserted
pShapeLines - pointer to an array of structure that describes shape lines
lNumShapeLines - number of the shape lines the array in pShapeLines

Return

FALSE - if error occurred

MDCS_ASSAY_GetMeasurementAttributes

```
BOOL  MDCS_ASSAY_GetMeasurementAttributes(
HDBHANDLE hHandle,
LONGLONG lAssayID,
LONGLONG lMeasurementID,
MDCS_ST_COLUMNPROP* pAssMeas,
BOOL bInsert = FALSE
);
```

Purpose

Function to get a measurement description based on measurement set ID and assay ID.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle
lAssayID - assay ID
lMeasurementID - measurement ID
bInsert - if TRUE, create a new record if it does not exist

Output

pAssMeas - structure that contains results of the query

Return

FALSE - if error occurred

MDCS_ASSAY_GetAssayMeasurementRecord

```
BOOL  MDCS_ASSAY_GetAssayMeasurementRecord(
HDBHANDLE hHandle,
LONGLONG lAssayID,
LONGLONG lMeasurementID,
MDCS_GetDBResultsCCallback* pResultCallback
);
```

Purpose

Function to get a measurements description based on measurement set ID and assay ID.

Parameters

hHandle - database connection handle
lAssayID - assay ID
lMeasurementID - measurement ID
pResultCallback - pointer to callback function to get data

Output

Columns from the ASSAYS and Table_COLUMNS table

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_GetAssayMeasurementsByPlateAndMeasurement

```
BOOL
MDCS_ASSAY_GetAssayMeasurementsByPlateAndMeasurement(
HDBHANDLE hHandle,
LONGLONG lPlateID,
LONGLONG* plMeasurementID,
MDCS_GetDBResultsCCallback* pResultCallback
);
```

Purpose

To get measurement set measurement by plate ID and measurement ID.

Parameters

hHandle - database connection handle
lPlateID - plate ID
plMeasurementID - measurement ID of global measurement
pResultCallback - pointer to callback function to get data

Output

Columns:
TABLE_ID - table name
COLUMN_NAME
ASSAY_ID
COLUMN_TYPE
COLUMN_NAME_EXT

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_GetMeasurementsBySiteID

```
BOOL  MDCS_ASSAY_GetMeasurementsBySiteID(
HDBHANDLE hHandle,
LONGLONG lSiteID,
MDCS_GetDBResultsCCallback* pResultCallback
);
```

Purpose

To get measurements by site ID.

Parameters

hHandle - database connection handle
lSiteID - site ID
pResultCallback - pointer to callback function to get data

Output

Columns:
TABLE_ID - table name
COLUMN_NAME
ASSAY_ID
COLUMN_TYPE
COLUMN_NAME_EXT

Return

FALSE - if error occurred

MDCS_ASSAY_GetMeasurementInfoByAssayAndColumnName

```
BOOL  MDCS_ASSAY_GetMeasurementInfoByAssayAndColumnName(
HDBHANDLE hHandle,
LONGLONG lAssayID,
LPCSTR pszAttrDBColumnName,
MDCS_ST_COLUMNPROP& stAssayMeas
);
```

Purpose

Function to get a measurement description based on measurement set ID and database column name.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle

lAssayID - assay ID

pszAttrDBColumnName - structure describes the attribute column in the database

bInsert - if TRUE, create a new record if it does not exist

Output

stAssayMeas - structure that contains results of the query

Return

FALSE - if error occurred

MDCS_ASSAY_GetMeasurementByFunctionAndParameterName

```
BOOL
MDCS_ASSAY_GetMeasurementByFunctionAndParameterName(
HDBHANDLE hHandle,
LPCSTR pszFunctionName,
LPCSTR pszParameterName,
const MDCS_E_ColumnType& eColType,
MDCS_ST_COLUMNPROP& stAssayMeas
);
```

Purpose

Function to get a measurement attribute description based on parameter and functions names.

Parameters

hHandle - database connection handle

pszFunctionName - function name

pszParameterName - parameter name

eColType - data format

Output

stAssayMeas - structure that contains results of the query

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_GetMeasurementRecord

```
BOOL  MDCS_ASSAY_GetMeasurementRecord(
HDBHANDLE hHandle,
LONGLONG lColumnID,
MDCS_GetDBResultsCCallback* pResultCallback
);
```

Purpose

Function to get a data type record by its ID.

Parameters

hHandle - database connection handle
pResultCallback - pointer to a callback function to get data

Output

Returns records with columns:
ID - measurement ID
FUNCTION_NAME - function name
PARAMETER_NAME - parameter name
COLUMN_NAME_EXT - display name of column
COLUMN_NAME - data type name as in DATABASE
COLUMN_DESCRIPTION - data type description
COLUMN_TYPE - column type
TABLE_ID - name of the table where can be found
ENTITY_ID - measurement set ID

Return

FALSE - if error occurred

MDCS_ASSAY_Create

```
BOOL  MDCS_ASSAY_Create(
HDBHANDLE hHandle,
const MDCS_ST_Assay* pAssayIn,
MDCS_ST_Assay* pAssayOut,
LONGLONG lFolderID
);
```

Purpose

Function to create a new measurement set.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle

pAssayIn - pointer to a measurement set structure that will be used to create new measurement set

lFolderID - ID of the folder where measurement set will be created, if 0 - will create measurement set in the user's default folder

Output

pAssayOut - created measurement set

Return

FALSE - if error occurred

MDCS_ASSAY_GetAssaySiblingFolders

```
BOOL  MDCS_ASSAY_GetAssaySiblingFolders(
HDBHANDLE hHandle,
LONGLONG lParentFolderID,
MDCS_GetDBResultsCCallback* pResultCallback
);
```

Purpose

Function to get measurement set sibling folders.

Parameters

hHandle - database connection handle
lParentFolderID - ID of parent to retrieve siblings of
pResultCallback - pointer to a callback function to get data

Output result columns will be

FOLDER_NAME - folder name
FOLDER_ID - folder ID

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_CreateMeasurement

```
BOOL  MDCS_ASSAY_CreateMeasurement(
HDBHANDLE hHandle,
const MDCS_ST_COLUMNPROP* pstMeasurementIn,
MDCS_ST_COLUMNPROP* pstMeasurementOut
);
```

Purpose

Function to create a new measurement set measurement.

Parameters

hHandle - database connection handle
pstMeasurementIn - measurement description

Output

pstMeasurementOut - measurement description

Note: Either szColumnNameFull or the pair of szFunctionName and szMeasName are required, column type is required.

Return

FALSE - if error occurred

MDCS_ASSAY_AddMeasurement

```
BOOL  MDCS_ASSAY_AddMeasurement(
HDBHANDLE hHandle,
LONGLONG lAssayID,
LONGLONG lMeasID,
MDCS_ST_COLUMNPROP* pCreatedMeasurementOut
);
```

Purpose

Function to add a new measurement to a measurement set.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle
lAssayID - assay ID for which to add measurement
lMeasID - measurement ID to add

Output

pCreatedMeasurementOut - pointer to structure that contains created measurement description

Return

FALSE - if error occurred

MDCS_ASSAY_GetMeasurementByID

```
BOOL  MDCS_ASSAY_GetMeasurementByID(
HDBHANDLE hHandle,
LONGLONG lMeasuremntID,
MDCS_ST_COLUMNPROP* pstMeasurement
);
```

Purpose

Function to get a description of a measurement set measurement.

Parameters

hHandle - database connection handle
lMeasuremntID - measurement ID

Output

pstMeasurement - measurement description

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_GetMeasurementByName

```
BOOL MDCS_ASSAY_GetMeasurementByName(
HDBHANDLE hHandle,
LPCSTR pszName,
MDCS_ST_COLUMNPROP* pstMeasurement
);
```

Purpose

To get a description of a measurement set measurement.

Parameters

hHandle - database connection handle
pszName - measurement name

Output

pstMeasurement - measurement description

Return

FALSE - if error occurred

MDCS_ASSAY_Delete

```
BOOL  MDCS_ASSAY_Delete(
HDBHANDLE hHandle,
LONGLONG lAssayID,
BOOL bCanRestore = FALSE
);
```

Purpose

Function to delete an assay.

Parameters

hHandle - database connection handle
lAssayID - assay that needs to be deleted
bCanRestore - flag if TRUE, indicates that assay can be restored from Recycle bin

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_DeleteAllForPlate

```
BOOL  MDCS_ASSAY_DeleteAllForPlate(
HDBHANDLE hHandle,
LONGLONG lPlateID
);
```

Purpose

Function to delete all assays for a plate.

Parameters

hHandle - database connection handle
lPlateID - plate ID

Return

FALSE - if error occurred

MDCS_ASSAY_DeleteDataForPlate

```
BOOL  MDCS_ASSAY_DeleteDataForPlate(
HDBHANDLE hHandle,
LONGLONG lAssayID,
LONGLONG lPlateID
);
```

Purpose

Function to delete assay results for a plate.

Parameters

hHandle - database connection handle
lPlateID - plate ID
lAssayID - assay ID

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_DeleteFolder

```
BOOL  MDCS_ASSAY_DeleteFolder(
HDBHANDLE hHandle,
LONGLONG lFolderID
);
```

Purpose

Function to delete an assay folder.

Parameters

hHandle - database connection handle
lFolderID - folder ID in measurement sets tree

Return

FALSE - if error occurred

MDCS_ASSAY_CreateOutlinesTable

```
BOOL  MDCS_ASSAY_CreateOutlinesTable(
HDBHANDLE hHandle,
LONGLONG lAssayID,
LPSTR pszTableName,
int nSize,
int* pnSizeOut
);
```

Purpose

Function to create an assay outlines table.

Parameters

hHandle - database connection handle
lAssayID - assay ID
nSize - size of table name

Output

pszTableName - name of the table that contains outlines
pnSizeOut - actual size the table name

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_ManageFolderSecurity

```
BOOL  MDCS_ASSAY_ManageFolderSecurity(
HDBHANDLE hHandle,
LONGLONG lFolderID,
HWND hWnd = NULL,
LPCSTR pszDlgTitle = NULL
);
```

Purpose

Calls a dialog to manage security access to the assay folders.

Parameters

hHandle - database connection handle
I*FolderID* - folder ID
pszDlgTitle - dialog title
hWnd - application window

Return

FALSE - if error occurred

MDCS_ASSAY_CanModifyFolder

```
BOOL  MDCS_ASSAY_CanModifyFolder(
HDBHANDLE hHandle,
LONGLONG lFolderID,
BOOL& bCanModify
);
```

Purpose

To check if a user can modify a folder.

Parameters

hHandle - database connection handle
lFolderID - folder ID

Output

bCanModify - if TRUE - user can modify the folder

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_CreateFolder

```
BOOL  MDCS_ASSAY_CreateFolder(
HDBHANDLE hHandle,
const MDCS_ST_FolderInfo& stInfoIn,
MDCS_ST_FolderInfo & stInfoOut
);
```

Purpose

To create a new assay folder in the database.

Parameters

hHandle - database connection handle
stInfoIn - structure describes input folder information.
stInfoOut - structure describes output (created) folder information

Return

FALSE - if error occurred

MDCS_ASSAY_ModifyFolder

```
BOOL  MDCS_ASSAY_ModifyFolder(
HDBHANDLE hHandle,
const MDCS_ST_FolderInfo& stInfoIn
);
```

Purpose

To modify an assay folder in the database.

Parameters

hHandle - database connection handle
stInfoIn - structure describes folder information.

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_DoesSubFolderExist

```
BOOL  MDCS_ASSAY_DoesSubFolderExist(
HDBHANDLE hHandle,
LONGLONG lFolderID,
LPCSTR pzName,
LONGLONG &lSubfolderID
);
```

Purpose

Check to see if an assay folder exists in the database.

Parameters

hHandle - database connection handle
pzName - folder ID

Output

lSubfolderID - subfolder ID, if 0 subfolder does not exist, otherwise - folder ID

Return

FALSE - if folder does not exist

MDCS_ASSAY_GetAllInFolder

```
BOOL  MDCS_ASSAY_GetAllInFolder(
HDBHANDLE hHandle,
LONGLONG lFolderID,
MDCS_GetDBResultsCCallback* pResultCallback
);
```

Purpose

To get assays with basic descriptions in a folder.

Parameters

hHandle - database connection handle
lFolderID - measurement set folder ID
pResultCallback - pointer to a callback function to get data

Output

Fields that are always present

ASSAY_ID, ASSAY_NAME, SETTINGS_NAME, TIME_CREATED (as time-out value), TABLE_ID (table name where results are stored),

SHAPE_TABLE_NAME (name of the table where shape results are stored)

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_GetFolderPath

```
BOOL  MDCS_ASSAY_GetFolderPath(
HDBHANDLE hHandle,
LONGLONG lFolderID,
AxString& strPath
);
```

Purpose

To get the path of a folder.

Parameters

hHandle - database connection handle
lFolderID - measurement set folder ID

Output

strPath - will contain the path

MDCS_ASSAY_Reindex

```
BOOL  MDCS_ASSAY_Reindex(
HDBHANDLE hHandle,
LONGLONG lAssayID
);
```

Purpose

To reindex assay measurements.

Parameters

hHandle - database connection handle
lAssayID - assay ID

MDCS_ASSAY_ReindexShapes

```
BOOL  MDCS_ASSAY_ReindexShapes(
HDBHANDLE hHandle,
LONGLONG lAssayID
);
```

Purpose

To reindex assay cell shapes.

Parameters

Handle - database connection handle
lAssayID - assay folder ID

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_OptimizeAll

```
BOOL  MDCS_ASSAY_OptimizeAll(
HDBHANDLE hHandle,
MDCS_ProgressCallback * pCallBack
);
```

Purpose

To optimize all assay tables.

Parameters

hHandle - database connection handle
pCallBack - progress callback

Return

FALSE - if error occurred

MDCS_ASSAY_Copy

```
BOOL  MDCS_ASSAY_Copy(
HDBHANDLE hHandle,
LONGLONG lAssayIDSource,
LONGLONG lAssayIDDestination,
MDCS_ProgressCallback * pCallBack
);
```

Purpose

Function to copy an assay.

Parameters

hHandle - database connection handle
lAssayIDSource - source assay ID
lAssayIDDestination - destination assay ID
pCallBack - progress callback

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_Merge

```
BOOL  MDCS_ASSAY_Merge(
HDBHANDLE hHandle,
LONGLONG lAssayIDSource,
LONGLONG lAssayIDDestination,
BOOL bCopyObjects,
MDCS_ProgressCallback * pCallBack
);
```

Purpose

Function to merge assays.

Parameters

hHandle - database connection handle
lAssayIDSource - source assay ID
lAssayIDDestination - destination assay ID
bCopyObjects - option indicates that attachments should be copied
pCallBack - progress callback

Return

FALSE - if error occurred

MDCS_ASSAY_GetSiteCount

```
BOOL  MDCS_ASSAY_GetSiteCount(
HDBHANDLE hHandle,
LPCSTR pszTableName,
LONGLONG lSiteID,
LONGLONG& lSiteCount
);
```

Purpose

Function to count site appearance in the assay table.

Parameters

hHandle - database connection handle
lSiteID - site ID to look for
pszTableName - name of a table that contains outlines

Output

lSiteCount - number of times site appears in the assay table

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_GetSiteInfoImageByID

```
BOOL  MDCS_ASSAY_GetSiteInfoImageByID(
HDBHANDLE hHandle,
LONGLONG lImageID,
MDCS_ST_SiteImageInfo& stInfoOut
);
```

Purpose

Function to get information about the image using the image ID.

Parameters

hHandle - database connection handle
lImageID - image ID

Output

StInfoOut - structure of site information

Return

FALSE - if error occurred

MDCS_ASSAY_GetSiteInfoImageBySiteAndSeriesID

```
BOOL  MDCS_ASSAY_GetSiteInfoImageBySiteAndSeriesID(
HDBHANDLE hHandle,
LONGLONG lSiteID,
LONGLONG lSeriesID,
MDCS_ST_SiteImageInfo& stInfoOut
);
```

Purpose

Function to get information about the image using the Site and Series ID.

Parameters

hHandle - database connection handle
lSiteID - Site ID
lSeriesID - series ID

Output

StInfoOut - structure of Site information

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_MeasurementGetSiteCount

```
BOOL  MDCS_ASSAY_MeasurementGetSiteCount(
HDBHANDLE hHandle,
LONGLONG lAssayID,
LONGLONG lSiteID,
LONGLONG& lSiteCount
);
```

Purpose

Function to count site appearance in an assay.

Parameters

hHandle - database connection handle
lAssayID - assay ID
lSiteID - site ID

Output

lSiteCount - number of times site appears in the assay table

Return

FALSE - if error occurred

MDCS_ASSAY_DeleteMeasurement

```
BOOL  MDCS_ASSAY_DeleteMeasurement(
HDBHANDLE hHandle,
LONGLONG lMeasurementID
);
```

Purpose

Function to delete a measurement.

Parameters

hHandle - database connection handle
lMeasurementID - measurement ID

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_CellOutlinesGetSiteCount

```
BOOL  MDCS_ASSAY_CellOutlinesGetSiteCount(
HDBHANDLE hHandle,
LONGLONG lAssayID,
LONGLONG lSiteID,
LONGLONG& lSiteCount
);
```

Purpose

Function to count site appearance in assay cell outlines.

Parameters

hHandle - database connection handle
lAssayID - assay ID
lSiteID - site ID

Output

lSiteCount - number of times site appears in the assay table

Return

FALSE - if error occurred

MDCS_ASSAY_CreateNewName

```
BOOL  MDCS_ASSAY_CreateNewName(
HDBHANDLE hHandle,
LONGLONG lFolderID,
LPCSTR pszOrigName,
AxString& strNewName,
BOOL bCreateAsCopy
);
```

Purpose

Function to create a new assay name.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle

lFolderID - destination folder ID

pszOrigName - to check availability or create a new assay name in destination folder

strNewName - to check availability or create a new assay name in destination folder

bCreateAsCopy -if TRUE - the name is created using the prefix "Copy of <orig name>", if the name exists

Return

FALSE - if error occurred

MDCS_ASSAY_CreateRun

```
BOOL  MDCS_ASSAY_CreateRun(
HDBHANDLE hHandle,
const MDCS_ST_AssayRun& stAssayRun,
LONGLONG* plAssayRunID
);
```

Purpose

To create a new assay run record.

Parameters

hHandle - database connection handle

stAssayRun - assay run description

Output

plAssayRunID - generated assay runID

Return

FALSE - if error occurred

MDCS_ASSAY_GetLatestAssayRunID

```
BOOL  MDCS_ASSAY_GetLatestAssayRunID(
HDBHANDLE hHandle,
LONGLONG lAssayID,
LONGLONG* plRunID
);
```

Purpose

To get ID of the latest run on an assay.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle
lAssayID - assay ID

Output

plRunID - assay run ID

Return

FALSE - if error occurred

MDCS_ASSAY_GetByID

```
BOOL  MDCS_ASSAY_GetByID(
HDBHANDLE hHandle,
LONGLONG lAssayID,
MDCS_ST_Assay* pstAssay
);
```

Purpose

Function to get assay data by ID.

Parameters

hHandle - database connection handle
lAssayID - assay ID

Output

pstAssay - structure with assay information

Return

FALSE - if error occurred

MDCS_ASSAY_CreateProfile

```
BOOL  MDCS_ASSAY_CreateProfile(
HDBHANDLE hHandle,
const MDCS_ST_AssayProfile& stAssayProfile,
LONGLONG* lAssayProfileID
);
```

Purpose

To create a new assay profile record.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle
stAssayProfile - assay profile description

Output

lAssayProfileID - generated assay profile ID

Return

FALSE - if error occurred

MDCS_ASSAY_AssociateWithPlate

```
BOOL  MDCS_ASSAY_AssociateWithPlate(
HDBHANDLE hHandle,
LONGLONG lAssayID,
LONGLONG lPlateID,
BOOL* bRecordExisted,
LONGLONG* plPlateAssayId = NULL
);
```

Purpose

To associate an assay with a plate.

Parameters

hHandle - database connection handle
lAssayID - assay ID
lPlateID - plate ID

Output

bRecordExisted - if TRUE - indicates that record already existed in database
plPlateAssayId - assay plate ID

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_GetSpotID

```
BOOL  MDCS_ASSAY_GetSpotID(
HDBHANDLE hHandle,
LONGLONG lSpot,
LONGLONG lInstance,
LONGLONG lSeries,
LONGLONG lSite,
LONGLONG& lSpotID
);
```

Purpose

To get the spot ID from the instance, site and series.

Parameters

hHandle - database connection handle
lSpot - Measurements table number
lInstance - instance ID
lSite - site ID
lSeries - series ID

Output

lSpotID - Spot ID

Return

FALSE - if error occurred

MDCS_ASSAY_GetAllForPlate

```
BOOL  MDCS_ASSAY_GetAllForPlate(
HDBHANDLE hHandle,
LONGLONG lPlateID,
MDCS_GetDBResultsCCallback * pResultCallback
);
```

Purpose

Function to get all assays that are associated with a plate.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle
lPlateID - ID of a plate
pResultCallback - pointer to a callback function to get data

Output

Columns:
ASSAY_ID - assay ID
ASSAY_NAME - assay name
ASSAY_DESC - assay description
CREATOR_NAME - name of the user who created the assay

Return

FALSE - if error occurred

MDCS_ASSAY_GetProfiles

```
BOOL  MDCS_ASSAY_GetProfiles(
HDBHANDLE hHandle,
MDCS_GetDBResultsCCallback * pResultCallback
);
```

Purpose

Function to get all available profiles.

Parameters

hHandle - database connection handle
pResultCallback - pointer to a callback function to get data

Output

Columns:
ID - unique ID
NAME - record name
DESCRIPTION - record description
SETTINGS_NAME - name of the settings
OPERATOR - name of the user who created settings
DATA_STORAGE_TYPE - type of the data storage
SETTINGS_STORAGE_TYPE - type of a storage where the settings BLOB is stored (DATABASE or File Server)

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_GetProfile

```
BOOL  MDCS_ASSAY_GetProfile(
HDBHANDLE hHandle,
LONGLONG lProfileID,
MDCS_GetDBResultsCCallback * pResultCallback
);
```

Purpose

Function to get a profile.

Parameters

hHandle - database connection handle
lProfileID - profile ID
pResultCallback - pointer to a callback function to get data

Output

Columns:
ID - unique ID
NAME - record name
DESCRIPTION - record description
SETTINGS_NAME - name of the settings
OPERATOR - name of user who created settings
DATA_STORAGE_TYPE - type of the data storage
SETTINGS_STORAGE_TYPE - type of a storage where settings BLOB is stored (DATABASE or File Server)

Return

FALSE - if error occurred

MDCS_ASSAY_GetProfileInfo

```
BOOL  MDCS_ASSAY_GetProfileInfo(
HDBHANDLE hHandle,
LONGLONG lProfileID,
MDCS_ST_AssayProfile& stProfile
);
```

Purpose

Function to get profile information.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle
lProfileID - profile ID

Output

Returns columns:
stProfile - structure that contains profile information

Return

FALSE - if error occurred

MDCS_ASSAY_UpdateProfile

```
BOOL  MDCS_ASSAY_UpdateProfile(
HDBHANDLE hHandle,
const MDCS_ST_AssayProfile& stProfileToUpdate
);
```

Purpose

Function to update a profile record.

Parameters

hHandle - database connection handle
stProfileToUpdate - profile to update

Return

FALSE - if error occurred

MDCS_ASSAY_DeleteProfile

```
BOOL  MDCS_ASSAY_DeleteProfile(
HDBHANDLE hHandle,
LONGLONG lProfileID
);
```

Purpose

To delete an assay profile.

Parameters

hHandle - database connection handle
lProfileID - profile ID

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_GetRecord

```
BOOL  MDCS_ASSAY_GetRecord(
HDBHANDLE hHandle,
LONGLONG lAssayID,
MDCS_GetDBResultsCCallback * pResultCallback
);
```

Purpose

Function to get an assay record.

Parameters

hHandle - database connection handle
lAssayID - assay ID
pResultCallback - callback class

Output

Columns:
ID - unique ID
NAME - record name
DESCRIPTION - record description
SETTINGS_NAME - name of the settings
DATA_STORAGE_TYPE - type of the data storage
SETTINGS_STORAGE_TYPE - type of a storage where the settings BLOB is stored (DATABASE or File Server)

Return

FALSE - if error occurred

MDCS_ASSAY_GetAssayByRunID

```
BOOL  MDCS_ASSAY_GetAssayByRunID(
HDBHANDLE hHandle,
LONGLONG lRunID,
MDCS_GetDBResultsCCallback * pResultCallback
);
```

Purpose

Function to get a measurement set record using the RUN ID.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle
lRunID - run ID
pResultCallback - pointer to callback function to get data

Output

Columns:
ID - unique ID
NAME - record name
DESCRIPTION - record description
SETTINGS_NAME - name of the settings
DATA_STORAGE_TYPE - type of the data storage
SETTINGS_STORAGE_TYPE - type of a storage where the settings BLOB is stored (DATABASE or File Server)

Return

FALSE - if error occurred

MDCS_ASSAY_UsedInDatasets

```
BOOL  MDCS_ASSAY_UsedInDatasets(
HDBHANDLE hHandle,
LONGLONG lAssayID,
LONGLONG& lDatasets
);
```

Purpose

Function to check if the assay is used in datasets.

Parameters

hHandle - database connection handle
AssayID - assay ID

Output

lDatasets - number of datasets where the assay is used

Return

FALSE - if dialog cancelled.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_HavePermissionsToModify

```
BOOL  MDCS_ASSAY_HavePermissionsToModify(
HDBHANDLE hHandle,
LONGLONG lAssayID,
BOOL& bCanModify
);
```

Purpose

Function to check if the current user can modify an assay.

Parameters

hHandle - database connection handle
lAssayID - assay ID

Output

bCanModify - if TRUE - user can modify the assay

Return

FALSE - if dialog cancelled

MDCS_ASSAY_GetImageSourceRecords

```
BOOL  MDCS_ASSAY_GetImageSourceRecords(
HDBHANDLE hHandle,
MDCS_GetDBResultsCCallback* pResultCallback
);
```

Purpose

To get all records from the IMAGE_SOURCE table.

Parameters

hHandle - database connection handle
pResultCallback - pointer to callback function to get data

Output

Returns records with columns:
ID - image source ID
ACQ_INSTANCE_ID - acquisition instance ID
ASSAY_INSTANCE_ID - measurement set ID
SOURCE_DESCRIPTION - source description
SOURCE_ILLUMINATION - source illumination

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_GetStatisticalValuesForPlateAndAssay

```
BOOL  MDCS_ASSAY_GetStatisticalValuesForPlateAndAssay(
HDBHANDLE hHandle,
LONGLONG lPlateID,
LONGLONG lAssayID,
LONGLONG lMeasurementID,
const MDCS_E_Statistic& eStatistic,
BOOL bGroupSites,
BOOL bAllSites,
int nSiteX,
int nSiteY,
LPCSTR pszSeriesInfoIndex,
LPCSTR pszFilter,
MDCS_GetDBResultsCCallback * pResultCallback
);
```

Purpose

To get statistical values for the selected plate and assay.

Parameters

hHandle - database connection handle
pResultCallback - pointer to a callback function to get data
lPlateID - plate ID
lAssayID - assay ID
lMeasurementID - measurement ID
eStatistic - type of statistic to apply to data
bGroupSites - if TRUE, the data will be grouped by site
bAllSites - if TRUE, the data will be fetched for all series for a site
nSiteX - X position of a site
nSiteY - Y position of a site
pszSeriesInfoIndex - series information index
pszFilter - filter that should be applied to the data

Output

Contains columns with statistical values

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_GetStatisticalValuesForPlateAndMeasurement

```
BOOL
MDCS_ASSAY_GetStatisticalValuesForPlateAndMeasurement(
HDBHANDLE hHandle,
LONGLONG lPlateID,
LONGLONG lMeasurementID,
const MDCS_E_Statistic& eStatistic,
BOOL bGroupSites,
BOOL bAllSites,
int nSiteX,
int nSiteY,
LPCSTR pszSeriesInfoIndex,
LPCSTR pszFilter,
MDCS_GetDBResultsCCallback * pResultCallback
);
```

Purpose

To get statistical values for the selected plate and measurement.

Parameters

hHandle - database connection handle
pResultCallback - pointer to a callback function to get data
lPlateID - plate ID
lMeasurementID - measurement ID
eStatistic - type of statistic to apply to data
bGroupSites - if TRUE, the data is grouped by site
bAllSites - if TRUE, the data is fetched for all series for a site
nSiteX - X position of a site
nSiteY - Y position of a site
pszSeriesInfoIndex - series information index
pszFilter - filter that should be applied to the data

Output

Contains columns with statistical values

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_GetValuesForPlateAndAssay

```
BOOL  MDCS_ASSAY_GetValuesForPlateAndAssay(
HDBHANDLE hHandle,
LONGLONG lPlateID,
LONGLONG lAssayID,
LONGLONG lMeasurementID,
BOOL bSingleSite,
int nSiteX,
int nSiteY,
LPCSTR pszFilter,
MDCS_GetDBResultsCCallback * pResultCallback
);
```

Purpose

To get values for the selected plate and assay.

Parameters

hHandle - database connection handle
pResultCallback - pointer to a callback function to get data
lPlateID - plate ID
lAssayID - assay ID
lMeasurementID - measurement ID
bSingleSites - if TRUE, data will be fetched for single sites.
nSiteX - X position of a site
nSiteY - Y position of a site
pszFilter - filter that should be applied to the data

Output

Contains columns with statistical values

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_GetValuesForPlate

```
BOOL  MDCS_ASSAY_GetValuesForPlate(
HDBHANDLE hHandle,
LONGLONG lPlateID,
LONGLONG lMeasurementID,
BOOL bSingleSite,
int nSiteX,
int nSiteY,
LPCSTR pszFilter,
MDCS_GetDBResultsCCallback * pResultCallback
);
```

Purpose

To get values for the selected plate and measurement.

Parameters

hHandle - database connection handle
pResultCallback - pointer to a callback function to get data
lPlateID - plate ID
lMeasurementID - measurement ID
bSingleSites - if TRUE, data will be fetched for single sites
nSiteX - X position of a site
nSiteY - Y position of a site
pszFilter - filter that should be applied to data

Output

Includes all values for a plate (including WELL_X, WELL_Y, INSTANCE, SERIES_INFO_ID, SITE_ID)

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_GetAllSiteMeasurements

```
BOOL  MDCS_ASSAY_GetAllSiteMeasurements(
HDBHANDLE hHandle,
LONGLONG lAssayID,
LONGLONG lSiteID,
BOOL bNumeric,
MDCS_GetDBResultsCCallback * pResultCallback
);
```

Purpose

To get all measurements of a selected site.

Parameters

lAssayID - Assay ID
hHandle - database connection handle
pResultCallback - pointer to callback function to get data
lSiteID - site ID
bNumeric - if TRUE - fetch all numeric columns

Output

Contains site measurement columns from the TABLE_COLUMNS table

Return

FALSE - if error occurred

MDCS_ASSAY_GetMeasurementInfoByAssayAndPlate

```
BOOL  MDCS_ASSAY_GetMeasurementInfoByAssayAndPlate(
HDBHANDLE hHandle,
LONGLONG lAssayID,
LONGLONG lPlateID,
MDCS_GetDBResultsCCallback* pResultCallback
);
```

Purpose

To get data types records by plate and assay.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle
lAssayID - assay ID
lPlateID - Plate ID
pResultCallback - pointer to a callback function to get data

Output

Returns columns:
TABLE_ID - (table name)
COLUMN_NAME
ASSAY_ID
COLUMN_TYPE
COLUMN_NAME_EXTN (column type - S for string, N for numeric)
FUNCTION_NAME,
PARAMETER_NAME

Return

FALSE - if error occurred

MDCS_ASSAY_GetAssayInfoByNamePlateAndSettings

```
BOOL  MDCS_ASSAY_GetAssayInfoByNamePlateAndSettings(
HDBHANDLE hHandle,
LONGLONG lPlateID,
LPCSTR pszAssayName,
LPCSTR pszSettingsName,
MDCS_ST_Assay& stAssayOut
);
```

Purpose

To get assay information by Plate ID, Assay name and settings.

Parameters

hHandle - database connection handle
lPlateID - plate ID
pszAssayName - assay name
pszSettingsName - settings name

Output

stAssayOut - structure that contains assay information

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_GetAssayIDsOfPlate

```
BOOL  MDCS_ASSAY_GetAssayIDsOfPlate(
HDBHANDLE hHandle,
LONGLONG lPlateID,
MDCS_GetDBResultsCCallback * pResultCallback
);
```

Purpose

To get assay IDs that are available for a plate.

Parameters

hHandle - database connection handle
lPlateID - plate ID
pResultCallback - pointer to callback function to get data

Output

Assay IDs for the plate

Return

FALSE - if error occurred

MDCS_ASSAY_GetAssaySettingsOfPlate

```
BOOL  MDCS_ASSAY_GetAssaySettingsOfPlate(
HDBHANDLE hHandle,
LONGLONG lPlateID,
MDCS_GetDBResultsCCallback * pResultCallback
);
```

Purpose

To get unique measurement set settings of the plate.

Parameters

hHandle - database connection handle
lPlateID - plate ID
pResultCallback - pointer to callback function to get data

Output

Results set with fields:
ASSAY_ID
ASSAY_NAME
SETTINGS_NAME

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_GetMeasurementStatistic

```
BOOL  MDCS_ASSAY_GetMeasurementStatistic(
HDBHANDLE hHandle,
LONGLONG lMeasurementID,
MDCS_ST_MeasurementUseStatistic& stMeasurementSta
);
```

Purpose

To get the number of datasets and the number of measurement sets that use a measurement.

Parameters

hHandle - database connection handle
MeasurementID – measurement ID

Output

stMeasurementSta – structure that contains statistic information about datasets and measurement sets that use the measurement

Return

FALSE - if error occurred

MDCS_ASSAY_GetShapeLinesBySite

```
BOOL  MDCS_ASSAY_GetShapeLinesBySite (
HDBHANDLE hHandle,
LPCSTR pszTableName,
LONGLONG lSiteID,
MDCS_GetBlobColumnResults* pResultCallback,
LONGLONG lMaxSize = -1
);
```

Purpose

To get shape lines BLOB description per site.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle
pszTableName – table name
lSiteID – site id
lMaxSize – max size
pResultCallback - pointer to callback function to get data

Output

Result set with fields:
SITE_ID
SERIES_ID
INSTANCE
SHAPE_TYPE
LINES_TOTAL - total lines count for the shapes per combination
RECORD_COUNT - total records found for the shapes per combination above
BLOB_SIZE - data size to retrieve for the shapes per combination above

Return

FALSE - if error occurred

MDCS_ASSAY_GetShapeLinesBySiteAndSeries

```
BOOL  MDCS_ASSAY_GetShapeLinesBySiteAndSeries(
HDBHANDLE hHandle,
LPCSTR pszTableName,
LONGLONG lSiteID,
LONGLONG lSeriesID,
MDCS_GetBlobColumnResults* pResultCallback,
LONGLONG lMaxSize = -1);
```

Purpose

To get shape lines BLOB description per site by using site and series id

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle

pszTableName – table name

lSiteID – site id

lSeriesID – series id

lMaxSize – max size

pResultCallback - pointer to callback function to get data

Output

Columns:

SITE_ID - site id

SERIES_ID

INSTANCE

SHAPE_TYPE

LINES_TOTAL - total lines count for the shapes per combination
RECORD_COUNT - total records found for the shapes per combination above

BLOB_SIZE - data size to retrieve for the shapes per combination

Return

FALSE - if error occurred

MDCS_ASSAY_GetMarkedAssaysWithCallback

```
BOOL  MDCS_ASSAY_GetMarkedAssaysWithCallback (
HDBHANDLE hHandle,
MDCS_GetDBResultsCCallback* pResultCallback,
BOOL bMarkPerm = TRUE,
BOOL bWithSecurity = FALSE);
```

Purpose

To get deleted assays using callback

Parameters

hHandle - database connection handle

bMarkPerm – flag to indicate to get permanently deleted assay or restorable assay

bWithSecurity – flag to indicate whether to include plate security when getting assay info

pResultCallback - pointer to callback function to get data

Output

Output structures that contains results of the query

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_Restore

```
BOOL  MDCS_ASSAY_GetMarkedAssaysWithCallback (
HDBHANDLE hHandle,
      LONGLONG lAssayID);
```

Purpose

To restore a deleted assay from the recycle bin.

Parameters

hHandle - database connection handle
lAssayID – assay ID

Output

None

Return

FALSE - if error occurred

MDCS_ASSAY_CreateForPlate

```
BOOL  MDCS_ASSAY_CreateForPlate(
HDBHANDLE hHandle,
      const MDCS_ST_Assay* pAssayIn,
MDCS_ST_Assay* pAssayOut,
LONGLONG lPlateID);
```

Purpose

To create an assay for a plate.

Parameters

hHandle - database connection handle
pAssayIn – a structure that describes assay informations (excluding the assay id)
lPlateID – plate id associated with the assay

Output

pAssayOut – a structure that describe assay information about the newly created assay (including the assay id)

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_GetMeasurementByName

```
BOOL  MDCS_ASSAY_GetMeasurementByName(
HDBHANDLE hHandle,
LPCSTR pszName,
MDCS_ST_COLUMNPROP* pstMeasurement,
LPCSTR pszScopeName = MDCS_cszMDCSCellMeasurementScope);
```

Purpose

To get a description of an assay measurement.

Parameters

hHandle - database connection handle
pszName – a name of the measurement
pszScopeName – Scope name of the measurement

Output

pstMeasurement – output structure that describes the measurement

Return

FALSE - if error occurred

MDCS_ASSAY_GetMeasurementByDBName

```
BOOL  MDCS_ASSAY_GetMeasurementByDBName (
HDBHANDLE hHandle,
LPCSTR pszColumnDBName,
MDCS_ST_COLUMNPROP& pstMeasurement,
LPCSTR pszScopeName = MDCS_cszMDCSCellMeasurementScope);
```

Purpose

To get a description of an assay measurement using the database name of the measurement.

Parameters

hHandle - database connection handle
pszColumnDBName – database name of the measurement
pszScopeName – Scope name of the measurement

Output

pstMeasurement – output structure that describes the measurement

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_GetScopeAttributeByName

```
BOOL  MDCS_ASSAY_GetScopeAttributeByName(
HDBHANDLE hHandle,
LPCSTR pszName,
MDCS_ST_ScopeAttribute& stAttr,
LPCSTR pszScopeName = MDCS_cszMDCSCellMeasurementScope);
```

Purpose

To get a description of a scope attribute using the display name.

Parameters

hHandle - database connection handle
pszName – display name of the attribute
pszScopeName – scope name of the attribute

Output

stAttr – output structure that describe the attribute

Return

FALSE - if error occurred

MDCS_ASSAY_GetScopeAttributeByID

```
BOOL  MDCS_ASSAY_GetScopeAttributeByID (
HDBHANDLE hHandle,
LPCSTR pszColumnID,
MDCS_ST_ScopeAttribute& stAttr,
LPCSTR pszScopeName = MDCS_cszMDCSCellMeasurementScope);
```

Purpose

To get a description of a scope attribute using the database name.

Parameters

hHandle - database connection handle
pszColumnID – a database name of the attribute
pszScopeName – scope name of the attribute

Output

stAttr – output structure that describes the attribute

Return

FALSE - if error occurred.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_DeleteMeasurementInAssays

```
BOOL  MDCS_ASSAY_DeleteMeasurementInAssays (
HDBHANDLE hHandle,
LPCSTR pszDBColumnName,
LONGLONG* arrAssayIDs,
INT_PTR nSize);
```

Purpose

To delete a measurement from all assays that use it but not the default. The default measurement is the measurement assay id = 0.

Parameters

hHandle - database connection handle
pszDBColumnName – the database name of the attribute
arrAssayIDs – array of assay IDs
nSize – size of arrAssayIDs

Output

NONE

Return

FALSE - if error occurred

MDCS_ASSAY_UpdateDataType

```
BOOL  MDCS_ASSAY_UpdateDataType (
HDBHANDLE hHandle,
LONGLONG lMeasurementID,
Const MDCS_ST_COLUMNPROP& stDataTypeNew);
```

Purpose

To update a measurement property.

Parameters

IMeasurementID -- a measurement ID

hHandle - database connection handle
IMeasurementID -- a measurement ID
stDataTypeNew – a structure of the database column used to replace the original

Output

NONE

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_UpdateDataTypeByAssay

```
BOOL  MDCS_ASSAY_UpdateDataTypeByAssay
HDBHANDLE hHandle,
LPCSTR stOrigDBCol,
MDCS_ST_COLUMNPROP& stDataTypeNew,
LONGLONG lAssayID);
```

Purpose

To modify a measurement for an assay based on the database column name and assay ID, but not the default measurement (the measurement where assay ID = 0).

Parameters

hHandle - database connection handle
stOrigDBCol - database column name
stDataTypeNew - structure of a measurement used to replace the original
lAssayID - assay ID

Output

NONE

Return

FALSE - if error occurred.

MDCS_ASSAY_CanModifyAssay

```
BOOL  MDCS_ASSAY_CanModifyAssay
HDBHANDLE hHandle,
LONGLONG lAssayID,
BOOL& bCanModify);
```

Purpose

To check if the current user can modify an assay.

Parameters

hHandle - database connection handle
lAssayID - assay ID

Output

bCanModify – if TRUE user can modify assay

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_CalculateStatisticResults

```
BOOL  MDCS_ASSAY_CalculateStatisticResults(
HDBHANDLE hHandle,
const LONGLONG* plarrAssayIDs,
int nNumElementArray,
const AxStringArray& axarrFilters,
MDCS_E_Statistic* arrApplyStatistic,
int nStatCount,
const MDCS_ST_ScopeAttribute* arrStatisticColumns,
int nStatColumnCount,
const MDCS_ST_ScopeAttribute* axarrGroupBy,
int nGroupByCount,
const MDCS_ST_ScopeAttribute* axarrOrderBy,
int nOrderByCount,
MDCS_GetDBResultsCCallback *pResultCallback,
BOOL bGroupByAssay = FALSE);
```

Purpose

To get statistic results on array of assay ids.

Parameters

hHandle - database connection handle
plarrAssayIDs - array of assay IDs
arrApplyStatistic- statistic that will be applied to columns in arrStatisticColumns
arrStatisticColumns - array of columns that will be used to calculate statistic
axarrGroupBy - array of group by columns
axarrOrderBy - array of order by columns
axarrFilters - array of filters
bGroupByAssay - if TRUE - identifier of assay will be added
pResultCallback - pointer to a callback function to get data
nNumElementArray - number of elements in plarrAssayIDs
nStatCount - number of elements in arrApplyStatistic
nStatColumnCount - number of elements in arrStatisticColumns

nGroupByCount - number of elements in axarrGroupBy
nOrderByCount - number of elements in axarrOrderBy

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Output

Recordset with fields:
Statistic data for arrStatisticColumns columns as *STATS_<column name>* and columns specified in axarrGroupBy and axarrOrderBy.

Columns:

in axarrOrderBy should match columns specified in axarrGroupBy

Return

FALSE - if error occurred.

MDCS_ASSAY_CalculateStatisticEx

```
BOOL  MDCS_ASSAY_CalculateStatisticEx(
HDBHANDLE hHandle,
const LONGLONG* plarrAssayIDs,
int nNumElementArray,
const AxStringArray& axarrFilters,
const MDCS_E_Statistic* arrApplyStatistic,
int nStatCount,
const MDCS_ST_ScopeAttribute* arrStatisticColumns,
int nStatColumns,
const MDCS_ST_ScopeAttribute* axarrGroupBy,
int nGroupByColumns,
const MDCS_ST_ScopeAttribute* axarrOrderBy,
int nOrderByColumns,
const MDCS_ST_ScopeAttribute* axarrInnerGroupBy,
int nInnerGroupBy,
const MDCS_E_Statistic* arrInnerStatistic,
int nInnerStatCount,

MDCS_GetDBResultsCCallback *pResultCallback,
BOOL bGroupByAssay = FALSE);
```

Purpose

To get statistic results on array of assay ids.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle

plarrAssayIDs - array of assay IDs

aarApplyStatistic - statistic that will be applied to columns in arrStatisticColumns

arrStatisticColumns - array of columns that will be used to calculate statistic

axarrGroupBy - array of group by columns

axarrOrderBy - array of order by columns

axarrFilters - array of filters

bGroupByAssay - if TRUE - identifier of assay will be added

arrInnerStatistic - statistic operation that will applied to internal query

nNumElementArray - number of elements in plarrAssayIDs

nStatCount - number of elements in arrApplyStatistic

nStatColumns - number of elements in arrStatisticColumns

nGroupByColumns - number of elements in axarrGroupBy

nOrderByColumns - number of elements in axarrOrderBy

nInnerGroupBy - number of elements in axarrInnerGroupBy

pResultCallback - pointer to a callback function to get data

nInnerStatCount - number of elements in arrInnerStatistic

axarrInnerGroupBy - array of group by columns for internal query

Output

Recordset with fields:

Statistic data for arrStatisticColumns columns as *STATS_<column name>columns* specified in axarrGroupBy

Columns:

in axarrOrderBy should match columns specified in axarrGroupBy

Return

FALSE - if error occurred.

MDCS_ASSAY_GetZPrime

```
BOOL  MDCS_ASSAY_GetZPrime (
HDBHANDLE hHandle,
LONGLONG lAssayID,
LPCSTR pszFilter,
LPCSTR pszMeasurementType,
MDCS_GetDBResultsCCallback *pResultCallback
);
```

Purpose

To calculate Z' on assay and measurement type.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle
lAssayID - assay ID
pszFilter - filter
pszMeasurementType - measurement type to use in calculation
pResultCallback - pointer to a callback function to get data

Output

Recordset with fields:
Z_PRIME - calculated Z'
PLATE_ID - plate ID

Return

FALSE - if error occurred

MDCS_ASSAY_GetZPrimeScopeAttribute

```
BOOL  MDCS_ASSAY_GetZPrimeScopeAttribute (
HDBHANDLE hHandle,
LONGLONG lAssayID,
LPCSTR pszFilter,
MDCS_ST_ScopeAttribute stMeasurement,
MDCS_GetDBResultsCCallback *pResultCallback
);
```

Purpose

To calculate Z' on the assay and scope attribute.

Parameters

hHandle - database connection handle
lAssayID - assay ID
pszFilter - filter
stMeasurement - measurement type to use in calculation
pResultCallback - pointer to a callback function to get data

Output

Recordset with fields:
Z_PRIME - calculated Z'
PLATE_ID - plate ID

Return

FALSE - if error occurred.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_GetUniqueMeasurementValues

```
BOOL  MDCS_ASSAY_GetUniqueMeasurementValues (
HDBHANDLE hHandle,
LONGLONG* larrAssayIDs,
const AxStringArray& sarrFilters,
const MDCS_ST_ScopeAttribute& stMeasurement,
MDCS_GetDBResultsCCallback *pResultCallback
);
```

Purpose

To get unique values of measurement type for array of assays.

Parameters

hHandle - database connection handle
larrAssayIDs - assay IDs
sarrFilters - array of assay filters
stMeasurement - measurement to use in calculation
pResultCallback - pointer to a callback function to get data

Output

Recordset with fields:
first column contains results
the name is equal to the database column name of the measurement..

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_FindMeasurementValues

```
BOOL  MDCS_ASSAY_ ASSAY_FindMeasurementValues (
HDBHANDLE hHandle,
LONGLONG lAssayID,
LPCSTR pszFilter,
const MDCS_ST_ScopeAttribute* pstMeasurement,
const AxStringArray& arrValues,
MDCS_ST_ScopeAttribute *paarResultsColumns,
int nArrSize,
MDCS_GetDBResultsCCallback *pResultCallback
);
```

Purpose

To finds values from provided array in a measurement set.

Parameters

hHandle - database connection handle
lAssayIDs - measurement set to search *on*
pszFilter - measurement filter
pstMeasurement - measurement column to search on
arrValues - array of values to be searched
nArrSize - number of elements in array
paarResultsColumns - array of result columns paarResultsColumns
pResultCallback - pointer to a callback function to get data

Output

Recordset with fields: first column contains results for the column from paarResultsColumn.

Return

FALSE - if error occurred.

MDCS_ASSAY_UpdateMeasurementData

```
BOOL  MDCS_ASSAY_UpdateMeasurementData (
HDBHANDLE hHandle,
LONGLONG lAssayID,
const MDCS_ST_Attribute& stColumnToUpdate,
LPCSTR pszTransformCrit,
LPCSTR pszWhereCrit = NULL
);
```

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Purpose

To update measurement data based on transformation criteria.

Parameters

hHandle - database connection handle
lAssayID - Assay ID
stColumnToUpdate - column to update
pszTransformCrit - transformation criteria (e.g. ((MDCS15 - 15) * 10)) measurement column to search on
pszWhereCrit - filtering criteria (for example, MDCS15 <> 0)

Output

NONE

Return

FALSE - if error occurred

MDCS_ASSAY_GetAllMSetUniqueAnnotation

```
BOOL  MDCS_ASSAY_GetAllMSetUniqueAnnotation (
HDBHANDLE hHandle,
MDCS_GetDBResultsCCallback* pResultCallback
);
```

Purpose

To get unique annotation of all measurement sets.

Parameters

hHandle - database connection handle
pResultCallback - pointer to a callback function to get data

Output

Columns:
COLUMN_ID - Database column name
COLUMN_NAME - Display name
COLUMN_TYPE - column datatype.

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_CreateAttribute

```
BOOL  MDCS_ASSAY_CreateAttribute (
HDBHANDLE hHandle,
const MDCS_ST_Attribute& stAttributeIn,
MDCS_ST_Attribute& stAttributeOut
);
```

Purpose

To create an Assay attribute.

Parameters

hHandle - database connection handle
stAttributeIn - attribute description

Output

stAttributeOut - attribute created

Return

FALSE - if error occurred

MDCS_ASSAY_GetAttributeInfoByDisplayName

```
BOOL MDCS_ASSAY_GetAttributeInfoByDisplayName (
HDBHANDLE hHandle,
LPCSTR pszDisplayName,
MDCS_ST_Attribute& stAttributeOut
);
```

Purpose

To get attribute information using its display name.

Parameters

hHandle - database connection handle
pszDisplayName - attribute display name

Output

stAttributeOut - structure that contains attribute information

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_GetAttributeInfoByDBName

```
BOOL  MDCS_ASSAY_GetAttributeInfoByDBName (
HDBHANDLE hHandle,
LPCSTR pszDBName,
MDCS_ST_Attribute& stAttributeOut
);
```

Purpose

To get attribute information using its internal name.

Parameters

hHandle - database connection handle
pszDBName - internal name of the attribute

Output

stAttributeOut - structure contains attribute info

Return

FALSE - if error occurred

MDCS_ASSAY_GetAttributeValueByDBName

```
BOOL  MDCS_ASSAY_GetAttributeValueByDBName (
HDBHANDLE hHandle,
LPCSTR pszDBName,
LONGLONG lAssayID,
AxString& strValue
);
```

Purpose

To get an attribute value using its internal name.

Parameters

hHandle - database connection handle
pszDBName - internal name of the attribute
lAssayID - assay id

Output

strValue - attribute value

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_GetAttributeValueByDisplayName

```
BOOL  MDCS_ASSAY_GetAttributeValueByDisplayName (
HDBHANDLE hHandle,
LPCSTR pszDisplayName,
LONGLONG lAssayID,
AxString& strValue
);
```

Purpose

To get an attribute value using its display name.

Parameters

hHandle - database connection handle
pszDisplayName - display name of the attribute
lAssayID - assay ID

Output

strValue - attribute value

RETURN

FALSE - if error occurred

MDCS_ASSAY_AssignAttributeValueString

```
BOOL  MDCS_ASSAY_AssignAttributeValueString (
HDBHANDLE hHandle,
LONGLONG lAssayID,
LPCSTR pszDBName,
LPCSTR pszValue
);
```

Purpose

To assign an assay attribute value using its internal name.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle
pszDBName - internal name of attribute
lAssayID - Assay ID of the assay to change the attribute value for
pszValue – value to assign to the attribute

Output

NONE

Return

FALSE - if error occurred

MDCS_ASSAY_AssignAttributeValueLong

```
BOOL  MDCS_ASSAY_AssignAttributeValueLong (
HDBHANDLE hHandle,
LONGLONG lAssayID,
LPCSTR pszDBName,
LONGLONG* plValue
);
```

Purpose

To assign an assay attribute value using its internal name.

Parameters

hHandle - database connection handle
pszDBName - internal name of the attribute
lAssayID - Assay ID of the assay to change the attribute value for
plValue – long value to assign to the attribute

Output

NONE

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_AssignAttributeValueFloat

```
BOOL  MDCS_ASSAY_AssignAttributeValueFloat (
HDBHANDLE hHandle,
LONGLONG lAssayID,
LPCSTR pszDBName,
float* pfValue
);
```

Purpose

To an assign assay attribute value using its internal name.

Parameters

hHandle - database connection handle
pszDBName - internal name of the attribute
lAssayID - Assay ID of the assay to change the attribute value for
pfValue – float value to assign to the attribute

OutPUt

NONE

Return

FALSE - if error occurred

MDCS_ASSAY_RenameAttribute

```
BOOL  MDCS_ASSAY_RenameAttribute (
HDBHANDLE hHandle,
const MDCS_ST_Attribute& stAttrribute,
LPCSTR pszNewName
);
```

Purpose

To rename an assay attribute.

Parameters

hHandle - database connection handle
stAttrribute - attribute to modify
pszNewName - new name of the attribute

Output

NONE

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_DeleteAttribute

```
BOOL  MDCS_ASSAY_DeleteAttribute (
HDBHANDLE hHandle,
LPCSTR pszDBName
);
```

Purpose

To delete an assay attribute using its internal name.

Parameters

hHandle - database connection handle
pszDBName – internal name of the attribute to be deleted

Output

NONE

Return

FALSE - if error occurred

MDCS_ASSAY_GetHeaderAndFileInfo

```
BOOL  MDCS_ASSAY_GetHeaderAndFileInfo (
HDBHANDLE hHandle,
LONGLONG lAssayID,
MDCS_GetDBResultsCCallback* pCallback
);
```

Purpose

To get header information and file import location of a measurement set.

Parameters

hHandle - database connection handle
lAssayID – Assay ID for the assay to get header and file information for
pResultCallback - pointer to a callback function to get data

Output

Return columns:
FILE_PATH - file path
FILE_COMPUTER_NAME - name of the computer where the file was imported from.
FILE_NAME - file name
HEADER_INFO - header file information

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_UpdateMeasurementSetName

```
BOOL  MDCS_ASSAY_UpdateMeasurementSetName (
HDBHANDLE hHandle,
LONGLONG lAssayID,
LPCSTR pszNewName
);
```

Purpose

To update the name of a measurement set.

Parameters

hHandle - database connection handle
lAssayID – Assay ID of the assay for which to update the name
pszNewName – new name

Output

NONE

Return

FALSE - if error occurred

MDCS_ASSAY_UpdateMeasurementSetDescription

```
BOOL  MDCS_ASSAY_UpdateMeasurementSetDescription(
HDBHANDLE hHandle,
LONGLONG lAssayID,
LPCSTR pszNewDesc
);
```

Purpose

To update the description of a measurement set.

Parameters

hHandle - database connection handle
lAssayID – Assay ID of the assay to update the description for
pszNewDesc – new description

Output

NONE

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAY_GetUniqueMeasurementValuesByPlate

```
BOOL  MDCS_ASSAY_GetUniqueMeasurementValuesByPlate (
HDBHANDLE hHandle,
LONGLONG lPlateID,
MDCS_ST_ScopeAttribute* arrMeasurements,
int nArraySize,
MDCS_GetDBResultsCCallback *pResultCallback
);
```

Purpose

To get unique values of measurement types for a plate.

Parameters

hHandle - database connection handle
lPlateID - plate ID
arrMeasurements - array of measurement data types
nArraySize - array size
pResultCallback - pointer to a callback function to get data

Output

Returns a recordset with fields corresponding to the measurement types in the array, ordered by the first column.

Return

FALSE - if error occurred

Note: Works only on cell attributes.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Common Database Functions

8

This chapter contains common database functions.

Table 8-1: Common Database Functions

Function Name	**Description**
MDCS_DATABASE_Optimize	To optimize the current database
MDCS_DATABASE_Compact	To compact a database. Available only for SQL server
MDCS_DATABASE_CountActiveConnections	To count active connections
MDCS_DATABASE_GetSize	To get the size of the database (in bytes). Available only for SQL server
MDCS_DATABASE_GetAvailableDatabases	To get available databases.
MDCS_DATABASE_GetVersion	To get the database version
MDCS_UTILS_RemoveMarkedData	To remove all data that was marked as deleted
MDCS_UTILS_RemoveMarkedDataEx	To remove all data that was marked as deleted
MDCS_UTILS_DropTable	To drop a table
MDCS_UTILS_DoesObjectExist	To check if an object exists in the database
MDCS_UTILS_Execute	To execute a statement that does not return any value
MDCS_UTILS_CreateTable	To create a table
MDCS_UTILS_AddColumnToTable	To add a column to a table
MDCS_UTILS_CreateForeignKey	To create a foreign key
MDCS_UTILS_CreateStoredProc	To create a stored procedure
MDCS_UPD_CreateHistoryRecord	To create a new history record
MDCS_UPD_UpdateHistoryRecordStatus	To update the update history status field
MDCS_UPD_GetHistoryRecord	To get a history record
MDCS_UPD_FindFinishedUpdates	To find a finished update
MDCS_UPD_UpdateDatabaseVersion	To update the DB_VERSIONS table

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_DATABASE_Optimize

```
BOOL  MDCS_DATABASE_Optimize(
HDBHANDLE hHandle,
MDCS_ProgressCallback * pCallBack
BOOL bQuickOpt = FALSE
);
```

Purpose

Function to optimize current database.

Parameters

hHandle - database connection handle
pCallBack - progress callback
bQuickOpt - if TRUE, performs the quick optimization that does not include any dynamic tables

Return

FALSE - if error occurred

MDCS_DATABASE_Compact

```
BOOL  MDCS_DATABASE_Compact(
HDBHANDLE hHandle,
LPCSTR pszDatabaseName,
UINT uPrecentLeave,
MDCS_ProgressCallback * pCallBack = NULL
);
```

Purpose

Function to compact a database. Available only for SQL Server.

Parameters

hHandle - database connection handle
pszDatabaseName - name of the database to compact
uPrecentLeave - percentage of free space that should be left after compacting
pCallback - progress callback

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_DATABASE_CountActiveConnections

```
BOOL  MDCS_DATABASE_CountActiveConnections(
HDBHANDLE hHandle,
LONGLONG& lConnectionFound,
LPCSTR pszDatabaseName = NULL
);
```

Purpose

Function to count active connections. Available only for SQL Server.

Parameters

hHandle - database connection handle
pszDatabaseName - name of the database

Output

lConnectionFound - number of connections found

Return

FALSE - if error occurred

MDCS_DATABASE_GetSize

```
BOOL  MDCS_DATABASE_GetSize(
HDBHANDLE hHandle,
LPCSTR pszDatabaseName,
LONGLONG& lSize
);
```

Purpose

Function to get the size of the database (in bytes). Available only for SQL Server.

Parameters

hHandle - database connection handle
pszDatabaseName - name of the database

Output

lSize - size of the database

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_DATABASE_GetAvailableDatabases

```
BOOL  MDCS_DATABASE_GetAvailableDatabases(
HDBHANDLE hHandle,
AxStringArray& arrDBs,
AxStringArray* parrFilterTables = NULL
);
```

Purpose

Function to get available databases.

Parameters

hHandle - database connection handle
paarFilterTables - tables that should exist in the database

Output

arrDBS - list of databases found on server

Return

FALSE - if error occurred

MDCS_DATABASE_GetVersion

```
BOOL  MDCS_DATABASE_GetVersion(
HDBHANDLE hHandle,
MDCS_ST_DBVersion& stDBVersion
);
```

Purpose

Function to get the database version.

Parameters

hHandle - database connection handle

Output

stDBVersion - database version

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_UTILS_RemoveMarkedData

```
BOOL  MDCS_UTILS_RemoveMarkedData(
HDBHANDLE hHandle,
MDCS_ProgressCallback * pCallBack
);
```

Purpose

Function to remove all data that was marked as deleted. Only members of admin group can run this function.

Parameters

hHandle - database connection handle
pCallBack - progress callback

Return

FALSE - if error occurred

MDCS_UTILS_RemoveMarkedDataEx

```
BOOL  MDCS_UTILS_RemoveMarkedDataEX(
HDBHANDLE hHandle,
MDCS_ProgressCallback * pCallBack
BOOL bRemovePlateByPlate = TRUE
);
```

Purpose

Function to remove all data that was marked as deleted.
Only members of the admin group can run this function.

Parameters

hHandle - database connection handle
pCallBack - progress callback
bRemovePlateByPlate - if TRUE, deleted plate by plate, instead of table by table

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_UTILS_DropTable

```
BOOL  MDCS_UTILS_DropTable(
HDBHANDLE hHandle,
LPCSTR pszTableName,
BOOL bTempSpace = FALSE
);
```

Purpose

Function to drop a table.

Parameters

hHandle - database connection handle
pszTableName - table name
bTempSpace - if TRUE, the table will be dropped from the temp space

Return

FALSE - if error occurred

MDCS_UTILS_DoesObjectExist

```
BOOL  MDCS_UTILS_DoesObjectExist (
HDBHANDLE hHandle,
LPCSTR pszObjName,
BOOL& bExists,
BOOL bObjTemp = FALSE
);
```

Purpose

Function to check if an object exists in database.

Parameters

hHandle -database connection handle
pszObjName - name of the object (table, view)
bObjTemp - if TRUE, the object is temporary

Output

bExists - TRUE, if object exists

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_UTILS_Execute

```
BOOL  MDCS_UTILS_Execute (
HDBHANDLE hHandle,
LPCSTR pszQuery
);
```

Purpose

Function to execute a statement that does not return any value.

Parameters

hHandle -database connection handle
pszQuery - query string

Return

FALSE - if error occurred

MDCS_UTILS_CreateTable

```
BOOL  MDCS_UTILS_CreateTable (
HDBHANDLE hHandle,
LPCSTR pszTableName,
const MDCS_ST_DBColun* arrColumns,
int nArraySize,
LPCTSTR pszPrefix = Null
);
```

Purpose

Function to create a table.

Parameters

hHandle -database connection handle
pszTableName - name of the table
nArraySize - size of array of columns
arrColumns - array of table columns
pszPrefix - table prefix

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_UTILS_AddColumnToTable

```
BOOL MDCS_UTILS_AddColumnToTable (
HDBHANDLE hHandle,
BOOL& bExists,
LPCSTR strTableName,
LPCSTR strColumnName,
const MDCS_E_ColumnType& eColumnType,
int nColumnLength = 0,
BOOL bNull = TRUE,
BOOL bTempTable = FALSE
);
```

Purpose

Function to add a column to a table.

Parameters

hHandle -database connection handle
strTableName - name of the table
strColumnName - name of the column
nColumnLength - length of the column
bNull - if TRUE - Null allowed
bTempTable - if TRUE, table is a temporary table
eColumnType - Column type

Return

FALSE - if error occurred

MDCS_UTILS_CreateForeignKey

```
BOOL MDCS_UTILS_CreateForeignKey (
HDBHANDLE hHandle,
LPCTSTR pszTableName,
LPCTSTR pszColumnName,
LPCTSTR pszIndexName,
LPCTSTR pszForeignKey,
LPCTSTR pszForeignKeyTable,
BOOL bTempTable=FALSE);
);
```

Purpose

Function to create a foreign key.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle -database connection handle
pszTableName - table name
pszColumnName - column name
pszIndexName - index name
pszForeignKey - foreign key column
pszForeignKeyTable - Foreign key table (table in which *pszForeignKey* is a primary key)
bTempTable - if TRUE, table is a temporary table

Return

FALSE - if error occurred

MDCS_UTILS_CreateStoredProc

```
BOOL MDCS_UTILS_CreateStoredProc (
HDBHANDLE hHandle,
LPCTSTR pszName,
LPCTSTR pszSQL
);
```

Purpose

Function to create a stored procedure.

Parameters

hHandle - database connection handle
pszName - stored procedure name
pszSQL - SQL string that defines the stored procedure

Return

FALSE - if error occurred

MDCS_UPD_CreateHistoryRecord

```
BOOL MDCS_UPD_CreateHistoryRecord (
HDBHANDLE hHandle,
const MDCS_ST_Update& stUpdate,
LONGLONG& lUpdateID
);
```

Purpose

Function to create a new history record.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle
stUpdate - History update structure
lUpdateID - update ID

Return

FALSE - if error occurred

MDCS_UPD_UpdateHistoryRecordStatus

```
BOOL MDCS_UPD_UpdateHistoryRecordStatus (
HDBHANDLE hHandle,
MDCS_ST_Update::MDCS_E_UpdateStatus eStatus,
LONGLONG lUHID
);
```

Purpose

Function to update the status field for a history record.

Parameters

hHandle - database connection handle
eStatus - status
lUHID - update history ID

Return

FALSE - if error occurred

MDCS_UPD_GetHistoryRecord

```
BOOL MDCS_UPD_GetHistoryRecord (
(HDBHANDLE hHandle,
LONGLONG lUHID,
MDCS_GetDBResultsCCallback* pResultCallback);
);
```

Purpose

Function to get a history record for an update history ID.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle
lUHID - update history ID
pResultCallback - pointer to callback function to get data

Output

The output history record will contain columns from the UPDATE_HISTORY table.

Return

FALSE - if error occurred

MDCS_UPD_FindFinishedUpdates

```
BOOL MDCS_UPD_FindFinishedUpdates
(HDBHANDLE hHandle,
LONGLONG lUHID,
MDCS_ST_Update::MDCS_E_SystemType eSType,
MDCS_GetDBResultsCCallback* pResultCallback)
```

Purpose

Function to find a finished update that matches the input update ID and type.

Parameters

hHandle - database connection handle
lUHID - update history ID
eSType - update type
pResultCallback - pointer to callback function to get data

Output

The output history record will contain columns from the UPDATE_HISTORY table.

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_UPD_UpdateDatabaseVersion

```
BOOL MDCS_UPD_UpdateDatabaseVersion(
HDBHANDLE hHandle,
LONGLONG lDBVerID,
LONGLONG lDBVerPara1,
LONGLONG lDBVerPara2,
LONGLONG lDBVerPara3
);
```

Purpose

Function to update the DB_Versions table.

Parameters

hHandle - database connection handle
lDBVerID - database version ID
lDBVerPara1 - database version leading number
lDBVerPara2 - database version middle number
lDBVerPara3 - database version trailing number

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Plate Functions

9

This chapter contains functions that you can use to work with plates.

Table 9-1: Plate Functions

Function Name	Description
MDCS_PLATE_CreatePlate	Create a new plate and set default access permissions.
MDCS_PLATE_GetInfo	To get plate information
MDCS_PLATE_GetAllPropertyAttributes	Will return description of columns for the plate attributes
MDCS_PLATE_GetAllOrderedByAttributes	Return all plates ordered by attributes
MDCS_PLATE_GetInfoBasedOnAssay	To get plate information for a measurement set
MDCS_PLATE_Delete	To delete a plate
MDCS_PLATE_DeleteImages	To delete plate images
MDCS_PLATE_GetUniqueAttributeValues	To get unique values for a plate attribute
MDCS_PLATE_GetAllByAttributes	To get plates based on specified attributes values
MDCS_PLATE_ManageSecurity	To call a dialog to manage security access for a plate
MDCS_PLATE_FolderManageSecurity	To call a dialog to manage security access for plates in a dynamic folder
MDCS_PLATE_ShareSiteImagesByTimeAndZIndex	To share images by site, timepoint and z index
MDCS_PLATE_CreateSite	To create a new site
MDCS_PLATE_CreateSeries	To create a new series
MDCS_PLATE_CreateImageSource	To create an image source
MDCS_PLATE_CreateImageRecord	To create an image record
MDCS_PLATE_ImportLayoutData	To create (import) a plate layout
MDCS_PLATE_ApplyLayoutToAssay	To apply a layout to an array
MDCS_PLATE_DeleteLayout	To delete a plate layout
MDCS_PLATE_GetAllRecords	To get all plate records
MDCS_PLATE_GetRecord	To get a single plate record
MDCS_PLATE_GetSiteRecord	To get a site record
MDCS_PLATE_GetSitesByPlate	To get all sites for a plate
MDCS_PLATE_GetImageRecord	To get an image record
MDCS_PLATE_GetImageRecordPerPlate	To get all image records for a plate
MDCS_PLATE_UpdateImageObjectID	To assign an image object ID to the plate image

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Table 9-1: Plate Functions (cont'd)

MDCS_PLATE_GetImageObjectIDForImage	To get the ID of the object record where the image is stored for the plate image
MDCS_PLATE_GetSeriesRecord	To get a series record by ID
MDCS_PLATE_GetSiteLocationsForPlate	To get all site positions for a plate
MDCS_PLATE_GetImageSourcesOfPlate	To get the image source records of a plate
MDCS_PLATE_GetMaxTimePointForPlate	To get the maximum the timepoint value for a plate
MDCS_PLATE_GetSeriesIDAtZAndT	To retrieve the series ID based on the site, Z index and T index
MDCS_PLATE_GetPlatesByDate	To get plates information by date range
MDCS_PLATE_UpdateAcquisition	To update an acquisition
MDCS_PLATE_CreateAttribute	To create a plate attribute
MDCS_PLATE_GetAttributeInfoByDisplayName	To get attribute information by display name
MDCS_PLATE_GetAttributeInfoByDBName	To get attribute information by database name
MDCS_PLATE_AssignAttributeValueString	To assign a plate attribute value(string)
MDCS_PLATE_AssignAttributeValueLong	To assign a plate attribute value(Long)
MDCS_PLATE_AssignAttributeValueFloat	To assign a plate attribute value(float)
MDCS_PLATE_UpdateInfo	To update plate information
MDCS_PLATE_GetInfoBasedOnAssaySet	To get plate information for a set of assays.
MDCS_PLATE_CreateLayout	To create a plate layout
MDCS_PLATE_ApplyLayoutToPlate	To apply a layout to a plate
MDCS_PLATE_GetTemplate	To get the annotation template using the id
MDCS_PLATE_GetTemplateByName	To get the annotation template using the name
MDCS_PLATE_GetTemplates	To get all plate annotation templates
MDCS_PLATE_UpdatePlateTemplate	To update plate annotation templates
MDCS_PLATE_RenameAttribute	To rename a plate attribute
MDCS_PLATE_CountPlateDatasets	To get the number of datasets where plate assays are used
MDCS_PLATE_GetAcqSiteCount	To get a count of all acquired sites per plate
MDCS_PLATE_GetAcqWellCount	To get a count of all acquired wells per plate
MDCS_PLATE_GetAcqSeriesCount	To get a count of all acquired series per plate

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Table 9-1: Plate Functions (cont'd)

MDCS_PLATE_GetCompoundCount	To get a count of compounds per plate
MDCS_PLATE_GetControlsCount	To get a count of controls per plate
MDCS_PLATE_GetControlStatistic	To get the control statistic per plate
MDCS_PLATE_GetCompleteImageInfo	To get complete image information for a plate
MDCS_PLATE_CanModify	To check if the current user can modify a plate
MDCS_PLATE_GetImageIDs	To get the IDs of plate images
MDCS_PLATE_GetThumbImageIDs	To get the IDs of plate thumbnails
MDCS_PLATE_ChangeStatus	To change the plate status

MDCS_PLATE_CreatePlate

```
BOOL  MDCS_PLATE_CreatePlate(
HDBHANDLE hHandle,
const MDCS_ST_PlateInfo& stPlateInfoIn,
MDCS_ST_PlateInfo* pstPlateInfoOut,
LONGLONG* plDefaultGroupID = NULL
);
```

Purpose

Function creates a new plate and sets default access permissions. Permissions are granted to administrators and the current user only.

Parameters

hHandle - database connection handle

stPlateInfoIn - plate description to insert

plDefaultGroupID - ID of a group that also has Lab Head permissions for the plate (If NULL - this will be ignored)

Output

pstPlateInfoOut - plate description as created in database

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_PLATE_GetInfo

```
BOOL  MDCS_PLATE_GetInfo(
HDBHANDLE hHandle,
LONGLONG lPlateID,
MDCS_ST_PlateInfo& stPlateInfoOut
);
```

Purpose

Function to get plate information.

Parameters

hHandle - database connection handle
lPlateID - plate ID

Output

pstPlateInfoOut - plate description as created in the database

Return

FALSE - if error occurred

MDCS_PLATE_GetAllPropertyAttributes

```
BOOL  MDCS_PLATE_GetAllPropertyAttributes(
HDBHANDLE hHandle,
MDCS_GetDBResultsCCallback* pResultCallback
);
```

Purpose

Function returns description of columns for the plate attributes.

Parameters

hHandle - database connection handle
pResultCallback - pointer to a callback function to get data

Output

Result columns are:

ATTR_DATABASE_NAME - name of the column that contains data for the attribute in the database, string

ATTR_NAME - display name of the attribute, string

ATTR_FORMAT - format of the data, integer cast to MDCS_E_ColumnType

ATTR_TYPE - type of the property, integer cast to MDCS_E_AttributeType

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_PLATE_GetAllOrderedByAttributes

```
BOOL  MDCS_PLATE_GetAllOrderedByAttributes(
HDBHANDLE hHandle,
const AxStringArray* pAXStringArray,
MDCS_GetDBResultsCCallback* pResultCallback
);
```

Purpose

Function returns all plates ordered by attributes (how the attributes are ordered in the array).

Parameters

hHandle - database connection handle
pAXStringArray - pointer to string array of property column names
pResultCallback - pointer to a callback function to get data

Output

Result columns are:
PLATE_ID - plate ID
PLATE_NAME -plate name
ACQ_ID - acquisition ID
GLOBAL_ID - plate global ID
CREATOR_ID creator ID
CREATOR_NAME - creator name
BARCODE - Plate barcode
TIME_CREATED - time when plate was created in seconds from 01/01/1970 00:00:00
ACQ_NAME - acquisition name

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_PLATE_GetInfoBasedOnAssay

```
BOOL  MDCS_PLATE_GetInfoBasedOnAssay(
HDBHANDLE hHandle,
LONGLONG lAssayID,
MDCS_ST_PlateInfo& stPlateInfoOut
);
```

Purpose

Function to get plate information for an assay.

Note: Plate information may not exist for a measurement set.

Parameters

hHandle - database connection handle
lAssayID - assay ID for which plate information will be retrieved

Output

stPlateInfoOut - plate information

Return

FALSE - if error occurred

MDCS_PLATE_Delete

```
BOOL  MDCS_PLATE_Delete(
HDBHANDLE hHandle,
LONGLONG lPlateID
);
```

Purpose

Function to delete a plate.

Note: Will not delete actual data; will only mark the data as deleted.

Parameters

hHandle - database connection handle
lPlateID - plate ID for the plate to be deleted

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_PLATE_DeleteImages

```
BOOL  MDCS_PLATE_DeleteImages(
HDBHANDLE hHandle,
LONGLONG lPlateID
);
```

Purpose

Function to delete plate images.

Note: Will not delete actual images; will only mark the images as deleted.

Parameters

hHandle - database connection handle
lPlateID - plate ID for the plate images to be deleted

Return

FALSE - if error occurred

MDCS_PLATE_GetUniqueAttributeValues

```
BOOL  MDCS_PLATE_GetUniqueAttributeValues(
HDBHANDLE hHandle,
LPCSTR pszAttrColumnName,
const AxStringArray* pAXAttrColumnNames,
const AxStringArray* pAXAttrValues,
AxStringArray* pAXValuesOut
);
```

Purpose

Function to get unique values for a plate attribute.

Parameters

hHandle - database connection handle
pszAttrColumnName - column that will contain unique values
pAXAttrColumnNames - attribute column names to query on
pAXAttrValues - values that attributes are equal to

Output

pAXValuesOut - array of unique values

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_PLATE_GetAllByAttributes

```
BOOL  MDCS_PLATE_GetAllByAttributes(
HDBHANDLE hHandle,
const AxStringArray* pAXDisplayColumns,
const AxStringArray* pAXAttrColumn,
const AxStringArray* pAXAttrValues,
MDCS_GetDBResultsCCallback* pResultCallback
);
```

Purpose

Function to get plates based on specified attribute values.

Parameters

HHandle - database connection handle
pAXDisplayColumns - columns to retrieve
pAXAttrColumn - columns to query on
pAXAttrValues - values corresponding to the columns to query on

Output

Result columns will be:
PLATE_ID - plate ID
PLATE_NAME - plate name
ACQ_ID - acquisition ID
GLOBAL_ID - plate global ID)
CREATOR_ID - plate creator ID
CREATOR_NAME - name of a person who created a plate (string)
BARCODE - plate barcode)
TIME_CREATED - time when plate was created in seconds from 01/01/1970 00:00:00
ACQ_NAME - acquisition name that was used to acquire plate and columns specified in pAXDisplayColumns

MDCS_PLATE_ManageSecurity

```
BOOL  MDCS_PLATE_ManageSecurity(
HDBHANDLE hHandle,
LONGLONG lPlateID,
HWND hWnd = NULL,
LPCSTR pszDlgTitle = NULL
);
```

Purpose

Calls a dialog to manage security access to the plate.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle
LPlateID - plate ID
pszDlgTitle - dialog title
hWnd - handle to the application window

Return

FALSE - if dialog cancelled

MDCS_PLATE_FolderManageSecurity

```
BOOL  MDCS_PLATE_FolderManageSecurity(
HDBHANDLE hHandle,
const AxStringArray& arrAttributes,
const AxStringArray& arrValues,
HWND hWnd = NULL,
LPCSTR pszDlgTitle = NULL
);
```

Purpose

Calls a dialog to manage security access for plates in a dynamic folder.

Parameters

hHandle - database connection handle
arrAttributes - array of attributes
arrValues - array of attribute values
pszDlgTitle - dialog title
hWnd - handle to the application window

Return

FALSE - if dialog cancelled

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_PLATE_ShareSiteImagesByTimeAndZIndex

```
BOOL  MDCS_PLATE_ShareSiteImagesByTimeAndZIndex(
HDBHANDLE hHandle,
LONGLONG lSiteID,
int nTIndexIn,
int nZIndexIn,
LONGLONG lSiteIDNew,
int nTIndexNew,
int nZIndexNew, LONGLONG& lCountUpdated,
LONGLONG lSelectBySourceID = 0
);
```

Purpose

To share images by site, timepoint and z index.

Parameters

hHandle - database connection handle
lSiteID - site ID to share
nTIndexIn - time index that should be copied
nZIndexIn - Z Index that should be copied
lSelectBySourceID - source ID, if needed to be part of selection (If 0 - ignored)
lSiteIDNew - destination site ID
nTIndexNew - destination time index
nZIndexNew - destination Z Index

Output

lCountUpdated - number of record copied

Return

FALSE - if it fails

MDCS_PLATE_CreateSite

```
BOOL  MDCS_PLATE_CreateSite(
HDBHANDLE hHandle,
const MDCS_ST_Site& stSite,
LONGLONG* plSiteID
);
```

Purpose

Function to create a new site.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle
stSite - site description

Output

plSiteID - site ID

Return

FALSE - if error occurred

MDCS_PLATE_CreateSeries

```
BOOL  MDCS_PLATE_CreateSeries(
HDBHANDLE hHandle,
const MDCS_ST_SeriesInfo& stSeries,
LONGLONG* plSeriesID
);
```

Purpose

Function to create a new series.

Parameters

hHandle - database connection handle
stSeries - series description

Output

plSeriesID - newly created series ID

Return

FALSE - if error occurred

MDCS_PLATE_CreateImageSource

```
BOOL  MDCS_PLATE_CreateImageSource(
HDBHANDLE hHandle,
const MDCS_ST_ImageSource& stImageSource,
LONGLONG* lImageSourceID
);
```

Purpose

Function to create image source.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle
stImageSource - image source description

Output

lImageSourceID - newly created Image source ID

Return

FALSE - if error occurred

MDCS_PLATE_CreateImageRecord

```
BOOL  MDCS_PLATE_CreateImageRecord (
HDBHANDLE hHandle,
const MDCS_ST_SiteImageInfo& stImageDesc,
LONGLONG* plRecordID
);
```

Purpose

Function to create an image record.

Parameters

hHandle - database connection handle
stImageDesc - image description

Output

plRecordID - newly created Image record ID

Return

FALSE - if error occurred

MDCS_PLATE_ImportLayoutData

```
BOOL  MDCS_PLATE_ImportLayoutData(
HDBHANDLE hHandle,
const MDCS_ST_PlateTemplate& stPlateTemplate,
MDCS_ImportPlateLayout* pDatasource,
LONGLONG* plLayoutID,
MDCS_ProgressCallback* pCallback
);
```

Purpose

Function to import a plate layout.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle
pDatasource - datasource that contains layout data
stPlateLayout - structure that describes plate layout
pCallback - callback class

Output

plLayoutID - newly created plate layout ID

Return

FALSE - if error occurred

MDCS_PLATE_ApplyLayoutToAssay

```
BOOL  MDCS_PLATE_ApplyLayoutToAssay(
HDBHANDLE hHandle,
LONGLONG lLayoutID,
LONGLONG lAssayID,
MDCS_ProgressCallback * pCallBack
);
```

Purpose

Function to apply a layout to an assay.

Parameters

hHandle - database connection handle
lLayoutID - layout ID
lAssayID - assay ID
pCallback - callback object

Return

FALSE - if error occurred

MDCS_PLATE_DeleteLayout

```
BOOL  MDCS_PLATE_DeleteLayout(
HDBHANDLE hHandle,
LONGLONG lLayoutID
);
```

Purpose

Function to delete plate layout.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle
lLayoutID - layout ID

Return

FALSE - if error occurred

MDCS_PLATE_GetAllRecords

```
BOOL  MDCS_PLATE_GetAllRecords(
HDBHANDLE hHandle,
MDCS_GetDBResultsCCallback* pResultCallback
);
```

Purpose

Function to get all plate records.

Parameters

hHandle - database connection handle
pResultCallback - pointer to a callback function to get data

Output

Returns records with columns
ID as plate ID
BATCH_ID as batch ID
X_WELLS, Y_WELLS number of wells
BARCODE - barcode
GLOBAL_ID - plate global ID
STORAGE_LOCATION - storage location
STORAGE_TYPE - storage type
PLATE_NAME - plate name
PLATE_DESCRIPTION - plate description
CREATOR - name of user who created the batch
TIME_CREATED - time when plate was created
NAME - plate name

Return

FALSE - if error occurred

Note: Storage locations always set to “database”; the DLL will decide where the images are located.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_PLATE_GetRecord

```
BOOL  MDCS_PLATE_GetRecord(
HDBHANDLE hHandle,
LONGLONG lPlateID,
MDCS_GetDBResultsCCallback* pResultCallback
);
```

Purpose

Function to get a single plate record.

Parameters

hHandle - database connection handle
lPlateID - plate ID
pResultCallback - pointer to a callback function to get data

Output

Returns records with columns:
ID as plate ID
BATCH_ID as batch ID
X_WELLS, Y_WELLS number of wells
BARCODE - barcode
GLOBAL_ID - plate global ID
STORAGE_LOCATION - storage location
STORAGE_TYPE - storage type
PLATE_NAME - plate name
PLATE_DESCRIPTION - plate description
CREATOR - name of user who create the batch
TIME_CREATED - time when plate was created
NAME - plate name

Return

FALSE - if error occurred

Note: Storage locations will always set to “database”, the DLL will decide where the images are located.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_PLATE_GetSiteRecord

```
BOOL  MDCS_PLATE_GetSiteRecord(
HDBHANDLE hHandle,
LONGLONG lSiteID,
MDCS_GetDBResultsCCallback* pResultCallback
);
```

Purpose

Function to get a site record.

Parameters

hHandle - database connection handle
lSiteID - site ID
pResultCallback - pointer to a callback function to get data

Output

Returns records with columns:
ID as plate ID
BATCH_ID as batch ID
X_WELLS, (number of wells), Y_WELLS (number of wells)
X_POSITION, (position within a well), Y_POSITION (position within a well)
TO_DELETE - (delete flag)

Return

FALSE - if error occurred

MDCS_PLATE_GetSitesByPlate

```
BOOL  MDCS_PLATE_GetSitesByPlate(
HDBHANDLE hHandle,
LONGLONG lPlateID,
MDCS_GetDBResultsCCallback* pResultCallback
);
```

Purpose

Function to get all sites for a plate.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle
lPlateID - plate ID
pResultCallback - pointer to a callback function to get data

Output

Returns records with columns:
ID as plate ID
BATCH_ID as batch ID
X_WELLS, (number of wells), Y_WELLS (number of wells)
X_POSITION, (position within a well), Y_POSITION (position within a well)
TO_DELETE - (delete flag)

Return

FALSE - if error occurred

MDCS_PLATE_GetImageRecord

```
BOOL  MDCS_PLATE_GetImageRecord(
HDBHANDLE hHandle,
LONGLONG lImageID,
MDCS_GetDBResultsCCallback* pResultCallback
);
```

Purpose

Function to get an image record.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle
lImageID - image ID
pResultCallback - pointer to a callback function to get data

Output

Returns records with columns:
ID as record ID
IMAGE_DATA_ID
SITE_ID
IMAGE_SOURCE_ID
ACQUIRE_DATE
TIMEPOINT
Z_POSITION
Z_STEP
SERIES_INFORMATION_ID
SOURCE_DESCRIPTION
STORAGE_LOCATION_NAME (location ID)
STORAGE_TYPE_NAME (always “DATABASE”)

Return

FALSE - if error occurred

MDCS_PLATE_GetImageRecordPerPlate

```
BOOL  MDCS_PLATE_GetImageRecordPerPlate(
HDBHANDLE hHandle,
LONGLONG lPlateID,
MDCS_GetDBResultsCCallback* pResultCallback
);
```

Purpose

Function to get all image records for a plate.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle
lPlateID - plate ID
pResultCallback - pointer to a callback function to get data

Output

Returns records with columns:
ID as record ID
IMAGE_DATA_ID
SITE_ID
IMAGE_SOURCE_ID
ACQUIRE_DATE
TIMEPOINT
Z_POSITION
Z_STEP
SERIES_INFORMATION_ID
SOURCE_DESCRIPTION
STORAGE_LOCATION_NAME (location ID)
STORAGE_TYPE_NAME (“DATABASE”)

Return

FALSE - if error occurred

MDCS_PLATE_UpdateImageObjectID

```
BOOL  MDCS_PLATE_UpdateImageObjectID(
HDBHANDLE hHandle,
LONGLONG lPlateImageID,
LONGLONG lImageObjectID
);
```

Purpose

Function to assign image object ID to the plate image.

Parameters

hHandle - database connection handle
lPlateImageID - plate image ID
lImageObjectID - image object ID

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_PLATE_GetImageObjectIDForImage

```
BOOL  MDCS_PLATE_GetImageObjectIDForImage(
HDBHANDLE hHandle,
LONGLONG lPlateImageID,
LONGLONG& lImageObjectID
);
```

Purpose

To get the image object ID (ID of the image record) for a plate image.

Parameters

hHandle - database connection handle
lPlateImageID - plate image ID

Output

lImageObjectID - image object ID

Return

FALSE - if error occurred

MDCS_PLATE_GetSeriesRecord

```
BOOL  MDCS_PLATE_GetSeriesRecord(
HDBHANDLE hHandle,
LONGLONG lSeriesID,
MDCS_GetDBResultsCCallback* pResultCallback
);
```

Purpose

Function to get a series record by ID.

Parameters

hHandle - database connection handle
lSeriesID - series ID
pResultCallback - pointer to a callback function to get data

Output

Returns records with columns:
ID as series ID
all other columns from the SERIES_INFO table

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_PLATE_GetSiteLocationsForPlate

```
BOOL  MDCS_PLATE_GetSiteLocationsForPlate(
HDBHANDLE hHandle,
LONGLONG lPlateID,
MDCS_GetDBResultsCCallback * pResultCallback
);
```

Purpose

Function to get all site positions for a plate.

Parameters

hHandle - database connection handle
lPlateID - Plate ID
pResultCallback - pointer to a callback function to get data

Output

Returns records with columns
X_POSITION, Y_POSITION values

Return

FALSE - if error occurred

MDCS_PLATE_GetImageSourcesOfPlate

```
BOOL  MDCS_PLATE_GetImageSourcesOfPlate(
HDBHANDLE hHandle,
LONGLONG lPlateID,
MDCS_GetDBResultsCCallback * pResultCallback
);
```

Purpose

To get image source records of a plate.

Parameters

hHandle - database connection handle
lPlateID - plate ID

Output columns

ID as image source ID
SOURCE_DESCRIPTION as source description
SOURCE_ILLUMINATION as source illumination

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_PLATE_GetMaxTimePointForPlate

```
BOOL  MDCS_PLATE_GetMaxTimePointForPlate(
HDBHANDLE hHandle,
LONGLONG lPlateID,
LONGLONG& lMaxTimepoint
);
```

Purpose

To get the maximum value for a timepoint for a plate.

Parameters

lPlateID - plate ID

Output

lMaxTimepoint - maximum timepoint value
hHandle - database connection handle

Return

FALSE - if error occurred

MDCS_PLATE_GetSeriesIDAtZAndT

```
MDCS_PLATE_GetSeriesIDAtZAndT(
HDBHANDLE hHandle,
LONGLONG lSiteID,
LONG lZIndex,
LONG lTIndex,
LONGLONG& lSeriesID
);
```

Purpose

To retrieve the series ID based on site, Z index and T index.

Parameters

hHandle - database connection handle
lSiteID - site ID
lZIndex - Z Index
lTIndex - Time Index

Output

lSeriesID - found series ID

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_PLATE_GetPlatesByDate

```
MDCS_PLATE_GetPlatesByDate(
HDBHANDLE hHandle,
LPCSTR pszStartDate,
LPCSTR pszEndDate,
MDCS_GetDBResultsCCallback * pResultCallback
);
```

Purpose

To get plate information by date range.

Parameters

hHandle - database connection handle
pszStartDate - start date
pszEndDate - end date
pResultCallback - pointer to a callback function to get data

Output

Result columns are:
PLATE_ID - plate ID
PLATE_NAME - plate name
ACQ_ID - acquisition ID
GLOBAL_ID - plate global ID
CREATOR_ID - creator ID
CREATOR_NAME - creator name
BARCODE - plate barcode
TIME_CREATED - time when plate was created in seconds from 01/01/1970 00:00:00

Return

FALSE - if error occurred

MDCS_PLATE_UpdateAcquisition

```
BOOL  MDCS_PLATE_UpdateAcquisition(
HDBHANDLE hHandle,
const MDCS_ST_Acquisition& stAcq
);
```

Purpose

To update an acquisition.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle
stAcq - acquisition description

Return

FALSE - if error occurred

MDCS_PLATE_CreateAttribute

```
BOOL  MDCS_PLATE_CreateAttribute(
HDBHANDLE hHandle,
const MDCS_ST_Attribute& stAttributeIn,
MDCS_ST_Attribute& stAttributeOut
);
```

Purpose

Function to create a plate attribute.

Parameters

hHandle - database connection handle
stAttributeIn - attribute description

Output

stAttributeOut - attribute created

Return

FALSE - if error occurred

MDCS_PLATE_GetAttributeInfoByDisplayName

```
BOOL  MDCS_PLATE_GetAttributeInfoByDisplayName(
HDBHANDLE hHandle,
LPCSTR pszDisplayName,
MDCS_ST_Attribute& stAttributeOut
);
```

Purpose

Function to get attribute information by display name.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle
pszDisplayName - display name of attribute

Output

stAttributeOut - attribute information

Return

FALSE - if error occurred

MDCS_PLATE_GetAttributeInfoByDBName

```
BOOL  MDCS_PLATE_GetAttributeInfoByDBName(
HDBHANDLE hHandle,
LPCSTR pszDBName,
MDCS_ST_Attribute& stAttributeOut
);
```

Purpose

Function to get attribute information by internal database name.

Parameters

hHandle - database connection handle
pszDBName - internal name of attribute

Output

stAttributeOut - attribute information

Return

FALSE - if error occurred

MDCS_PLATE_AssignAttributeValueString

```
BOOL  MDCS_PLATE_AssignAttributeValueString(
HDBHANDLE hHandle,
LONGLONG lPlateID,
LPCSTR pszDBName,
LPCSTR pszValue
);
```

Purpose

Function to assign a value to a plate attribute.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle
lPlateID - plate ID
pszDBName - internal name of attribute
pszValue - value to assign

Output

None

Return

FALSE - if error occurred

MDCS_PLATE_AssignAttributeValueLong

```
BOOL  MDCS_PLATE_AssignAttributeValueLong(
HDBHANDLE hHandle,
LONGLONG lPlateID,
LPCSTR pszDBName,
LONGLONG* plValue
);
```

Purpose

Function to assign plate attribute value.

Parameters

hHandle - database connection handle
lPlateID - ID of a plate that needs to change attribute value
pszDBName - internal name of attribute
(MDCS_ST_Attribute::szDBcolumnName)
plValue - value

Output

stAttributeOut - attribute created

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_PLATE_AssignAttributeValueFloat

```
BOOL  MDCS_PLATE_AssignAttributeValueFloat(
HDBHANDLE hHandle,
LONGLONG lPlateID,
LPCSTR pszDBName,
float* pfValue
);
```

Purpose

Function to assign a value to a plate attribute.

Parameters

hHandle - database connection handle
lPlateID - plate ID
pszDBName - internal name of attribute
pfValue - value to assign

Output

None

Return

FALSE - if error occurred

MDCS_PLATE_UpdateInfo

```
BOOL  MDCS_PLATE_UpdateInfo(
HDBHANDLE hHandle,
const MDCS_ST_PlateInfo& stInfoIn
);
```

Purpose

Function to update plate information.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle

stInfoIn - new information about the plate (plate ID must be specified)

Output

NONE

Return

FALSE - if error occurred

MDCS_PLATE_GetInfoBasedOnAssaySet

```
BOOL  MDCS_PLATE_GetInfoBasedOnAssaySet (
HDBHANDLE hHandle,
const LONGLONG* larrAssayID,
int nElementInArray,
MDCS_GetDBResultsCCallback * pResultCallback
);
```

Purpose

Function to get plate information for a set of assays.

Note: Plate information may not exist for an assay.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle
larrAssayID - array of assay IDs
nElementInArray - the number of elements in the array (larrAssayID)
pResultCallback - pointer to a callback function to get data

Output

Result columns will be:
ASSAY_ID - assay ID
PLATE_ID - plate ID
PLATE_NAME - plate name
ACQ_ID - acquisition ID
GLOBAL_ID - plate global ID
CREATOR_ID -creator ID
CREATOR_NAME - creator name
BARCODE - plate barcode
TIME_CREATED - time when plate was created in seconds from 01/01/1970 00:00:00)
ACQ_NAME - acquisition name

Return

FALSE - if error occurred

MDCS_PLATE_CreateLayout

```
BOOL  MDCS_PLATE_CreateLayout (
HDBHANDLE hHandle,
const MDCS_ST_PlateTemplate& stPlateTemplate,
MDCS_ST_PlateTemplate* pstPlateTemplateOut
);
```

Purpose

Function to create a plate layout.

Parameters

hHandle - database connection handle
stPlateLayout - structure that describes plate layout

Output

pstPlateTemplateOut - created layout

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_PLATE_ApplyLayoutToPlate

```
BOOL  MDCS_PLATE_ApplyLayoutToPlate (
HDBHANDLE hHandle,
LONGLONG lLayoutID,
LONGLONG lPlateID
);
```

Purpose

Function to apply a layout to a plate.

Parameters

hHandle - database connection handle
lLayoutID - layout ID
lPlateID - plate ID

Output

NONE

Return

FALSE - if error occurred

MDCS_PLATE_GetTemplate

```
BOOL  MDCS_PLATE_GetTemplate (
HDBHANDLE hHandle,
LONGLONG lTemplID,
MDCS_ST_PlateTemplate& stPlateTemplate
);
```

Purpose

Function to get the plate annotation template by template ID.

Parameters

hHandle - database connection handle
lTemplID - template ID

Output

stPlateTemplate - plate template

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_PLATE_GetTemplateByName

```
BOOL  MDCS_PLATE_GetTemplateByName (
HDBHANDLE hHandle,
LPCSTR pszTemplName,
MDCS_ST_PlateTemplate& stPlateTemplate
);
```

Purpose

Function to get the plate annotation template by providing its name.

Parameters

hHandle - database connection handle
pszTemplName - template name

Output

stPlateTemplate - plate template

Return

FALSE - if error occurred

MDCS_PLATE_GetTemplates

```
BOOL  MDCS_PLATE_GetTemplates (
HDBHANDLE hHandle,
MDCS_GetDBResultsCCallback * pResultCallback
);
```

Purpose

Function to get all existing plate annotation templates.

Parameters

hHandle - database connection handle
pResultCallback - pointer to a callback function to get data

Output

Callback will return columns described in MDCS_ST_PlateLayout structure

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_PLATE_UpdatePlateTemplate

```
BOOL  MDCS_PLATE_UpdatePlateTemplate (
HDBHANDLE hHandle,
const MDCS_ST_PlateTemplate& stPlateTemplateIn,
MDCS_ST_PlateTemplate* pstPlateTemplate
);
```

Purpose

Function to update a plate layout template.

Parameters

hHandle - database connection handle
stPlateTemplateIn - template input

Output

stPlateTemplate - updated plate template

Return

FALSE - if error occurred

MDCS_PLATE_RenameAttribute

```
BOOL  MDCS_PLATE_RenameAttribute (
HDBHANDLE hHandle,
const MDCS_ST_Attribute& stAttrribute,
LPCSTR pszNewName
);
```

Purpose

Function to rename a plate attribute.

Parameters

hHandle - database connection handle
stAttrribute - attribute to modify
pszNewName - new name of the attribute

Output

NONE

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_PLATE_CountPlateDatasets

```
BOOL  MDCS_PLATE_CountPlateDatasets (
HDBHANDLE hHandle,
LONGLONG lPlateID,
LONGLONG& lNumDataset
);
```

Purpose

Function to get the number of the datasets where a plate is used.

Parameters

hHandle - database connection handle
lPlateID - plate ID

Output

lNumDataset - number of the datasets where the plate is used

Return

FALSE - if error occurred

MDCS_PLATE_GetAcqSiteCount

```
BOOL  MDCS_PLATE_GetAcqSiteCount (
HDBHANDLE hHandle,
LONGLONG lPlateID,
LONGLONG& lCount
);
```

Purpose

Function to get the count of sites per plate..

Parameters

hHandle - database connection handle
lPlateID - plate ID

Output

lCount - number of sites per plate

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_PLATE_GetAcqWellCount

```
BOOL  MDCS_PLATE_GetAcqWellCount (
HDBHANDLE hHandle,
LONGLONG lPlateID,
LONGLONG& lCount
);
```

Purpose

Function to get the count of all acquired wells per plate.

Parameters

hHandle - database connection handle
lPlateID - plate ID

Output

lCount - number of acquired wells per plate

Return

FALSE - if error occurred

MDCS_PLATE_GetAcqSeriesCount

```
BOOL  MDCS_PLATE_GetAcqSeriesCount (
HDBHANDLE hHandle,
LONGLONG lPlateID,
LONGLONG& lCount
);
```

Purpose

Function to get the count of all acquired series per plate.

Parameters

hHandle - database connection handle
lPlateID - plate ID

Output

lCount - number of acquired series per plate

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_PLATE_GetCompoundCount

```
BOOL  MDCS_PLATE_GetCompoundCount (
HDBHANDLE hHandle,
LONGLONG lPlateID,
LONGLONG& lCount
);
```

Purpose

Function to get the count of all compounds per plate.

Parameters

hHandle - database connection handle
lPlateID - plate ID

Output

lCount - number of compounds per plate

Return

FALSE - if error occurred

MDCS_PLATE_GetControlsCount

```
BOOL  MDCS_PLATE_GetControlsCount (
HDBHANDLE hHandle,
LONGLONG lPlateID,
LONGLONG& lCount
);
```

Purpose

Function to get the count of controls per plate.

Parameters

hHandle - database connection handle
lPlateID - plate ID

Output

lCount - number of controls per plate

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_PLATE_GetControlStatistic

```
BOOL  MDCS_PLATE_GetControlStatistic (
HDBHANDLE hHandle,
LONGLONG lPlateID,
AxStringArray& arrControlNames,
AxStringArray& arrControlCount
);
```

Purpose

Function to get name and count of all controls in a plate.

Parameters

hHandle - database connection handle
lPlateID - plate ID

Output

arrControlNames - array that contains the control names
arrControlCount - array that contains the count of the corresponding control

Return

FALSE - if error occurred

MDCS_PLATE_GetCompleteImageInfo

```
BOOL MDCS_PLATE_GetCompleteImageInfo(
HDBHANDLE hHandle,
LONGLONG lPlateID,
MDCS_GetDBResultsCCallback* pResultCallback
);
```

Purpose

Function to get complete image information for a plate.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle
lPlateID - plate ID
pResultCallback - pointer to a callback function to get data

Output

Will return records with columns:
SITE_ID
SERIES_ID
IMAGE_SOURCE_ID,
OBJ_ID
IMAGE_ID
OBJ_SIZE
OBJ_SERVER_NAME
OBJ_EXT, THUMB_SIZE
THUMB_SERVER_NAME
LOCATION_ID,
THUMB_LOCATION_ID
WELL_X
WELL_Y
X_POSITION
Y_POSITION
Z_INDEX
Z_POSITION
T_INDEX
SOURCE_DESCRIPTION
SOURCE_ILLUMINATION

Return

FALSE - if error occurred

MDCS_PLATE_CanModify

```
BOOL  MDCS_PLATE_CanModify (
HDBHANDLE hHandle,
LONGLONG lPlateID,
BOOL* pbCan
);
```

Purpose

Function to check whether the current user can modify a plate.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle
lPlateID - plate ID

Output

pbCan - TRUE (if the plate can be modified*)*

Return

FALSE - if error occurred

MDCS_PLATE_GetImageIDs

```
BOOL MDCS_PLATE_GetImageIDs (
HDBHANDLE hHandle,
LONGLONG lPlateID,
AxStringArray& arrIDs
);
```

Purpose

Function to get the image IDs for plate images.

Parameters

hHandle - database connection handle
lPlateID - plate ID

Output

arrIDs - array of Image IDs found for the plate

Return

FALSE - if error occurred

MDCS_PLATE_GetThumbImageIDs

```
BOOL MDCS_PLATE_GetThumbImageIDs (
HDBHANDLE hHandle,
LONGLONG lPlateID,
AxStringArray& arrIDs
);
```

Purpose

Function to get the image IDS for plate thumbnails.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle
lPlateID - plate ID

Output

arrIDs - array of thumbnail image IDs found for the plate

MDCS_PLATE_ChangeStatus

```
BOOL MDCS_PLATE_ChangeStatus (
HDBHANDLE hHandle,
LONGLONG lPlateID,
MDCS_E_MarkStatus eStatus)
);
```

Purpose

Function to change the plate status.

Parameters

hHandle - database connection handle
lPlateID - plate ID
eStatus - plate status

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Acquisitions Functions

This chapter contains functions that you can use to work with acquisitions.

Table 10-1: Acquisitions Functions

Function Name	Description
MDCS_ACQUISITION_Delete	To delete an acquisition
MDCS_ACQUISITION_Create	Create a new acquisition
MDCS_ACQUISITION_CreateProfile	To create an acquisition profile
MDCS_ACQUISITION_GetProfileRecords	To get all records from the acquisition profile table
MDCS_ACQUISITION_GetProfile	To get a record from the acquisition profile table
MDCS_ACQUISITION_GetInstanceRecord	To get an acquisition instance record
MDCS_ACQUISITION_GetBatchRecords	To get all batch records
MDCS_ACQUISITION_GetBatchRecord	To get a batch record

MDCS_ACQUISITION_Delete

```
BOOL  MDCS_ACQUISITION_Delete(
HDBHANDLE hHandle,
LONGLONG lAcqID
);
```

Purpose

Function to delete an acquisition.

Parameters

hHandle - database connection handle
lAcqID - acquisition ID

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ACQUISITION_Create

```
BOOL  MDCS_ACQUISITION_Create(
HDBHANDLE hHandle,
const MDCS_ST_Acquisition& stAcq,
LONGLONG* plAcqID
);
```

Purpose

To create a new acquisition.

Parameters

hHandle - database connection handle
stAcq - acquisition description

Output

plAcqID - acquisition ID for newly created acquisition

Return

FALSE - if error occurred

MDCS_ACQUISITION_CreateProfile

```
BOOL  MDCS_ACQUISITION_CreateProfile(
HDBHANDLE hHandle,
const MDCS_ST_AcquisitionProfile& stAcqProf,
LONGLONG* plAcqProfID
);
```

Purpose

Function to create an acquisition profile.

Parameters

hHandle - database connection handle
stAcqProf - profile description

Output

plAcqProfID - Acquisition Profile ID for newly created profile

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ACQUISITION_GetProfileRecords

```
BOOL  MDCS_ACQUISITION_GetProfileRecords(
HDBHANDLE hHandle,
MDCS_GetDBResultsCCallback* pReslutCallback
);
```

Purpose

To get all records from the acquisition profile table.

Parameters

hHandle - database connection handle
pResultCallback - pointer to a callback function to get data

Output

Returns records with columns:
ID as ACQUISITION ID
NAME as acquisition name
DESCRIPTION as acquisition description

Return

FALSE - if error occurred

MDCS_ACQUISITION_GetProfile

```
BOOL  MDCS_ACQUISITION_GetProfile(
HDBHANDLE hHandle,
LONGLONG lProfileID,
MDCS_GetDBResultsCCallback* pResultCallback
);
```

Purpose

To get records for an acquisition profile.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle
lProfileID - profile ID
pResultCallback - pointer to a callback function to get data

Output

Returns records with columns:
ID as ACQUISITION ID
NAME as acquisition name
DESCRIPTION as acquisition description

Return

FALSE - if error occurred

MDCS_ACQUISITION_GetInstanceRecord

```
BOOL  MDCS_ACQUISITION_GetInstanceRecord(
HDBHANDLE hHandle,
LONGLONG lInstanceID,
MDCS_GetDBResultsCCallback* pResultCallback
);
```

Purpose

To get an acquisition instance record.

Parameters

hHandle - database connection handle
lInstanceID - Instance ID
pResultCallback - pointer to a callback function to get data

Output

Returns records with columns:
ID as ACQUISITION ID
NAME as acquisition name
DESCRIPTION as acquisition description
START_DATE - start date
END_DATE - end date

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ACQUISITION_GetBatchRecords

```
BOOL  MDCS_ACQUISITION_GetBatchRecords(
HDBHANDLE hHandle,
MDCS_GetDBResultsCCallback* pResultCallback
);
```

Purpose

To get all batch records.

Parameters

hHandle - database connection handle
pResultCallback - pointer to a callback function to get data

Output

Returns records with columns:
ID as ACQUISITION ID
ACQ_INSTANCE_ID as ACQUISITION ID
NAME as acquisition name
START_DATE - start date
END_DATE - end date
OPERATOR - name of the user who created batch
DESCRIPTION - batch description

Return

FALSE - if error occurred

MDCS_ACQUISITION_GetBatchRecord

```
BOOL  MDCS_ACQUISITION_GetBatchRecord(
HDBHANDLE hHandle,
LONGLONG lBatchID,
MDCS_GetDBResultsCCallback* pResultCallback
);
```

Purpose

To get a batch record.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle
lBatchId - batch ID

Output

Returns records with columns:
ID as ACQUISITION ID
ACQ_INSTANCE_ID as ACQUISITION ID
NAME as acquisition name
START_DATE - start date
END_DATE - end date
OPERATOR - name of user who created batch
DESCRIPTION - batch description

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Security Functions

11

This chapter contains functions that you can use to work with security.

Table 11-1: Security Functions

Function Name	Description
MDCS_SECURITY_ManageUsersDlg	To call a dialog to manage users
MDCS_SECURITY_ManageGroupsDlg	To call a dialog to manage groups
MDCS_SECURITY_GetUserGroups	To get a list of groups that the user has access to
MDCS_SECURITY_GetUserGroupsDlg	Display a selection dialog to choose groups
MDCS_SECURITY_GetUserInfo	To get user information
MDCS_SECURITY_IsUserInAdminGroup	To check if the user is in the Administrators group
MDCS_SECURITY_GetEveryoneGroupInfo	To get information about the everyone group
MDCS_SECURITY_ChangePassword	To change the password of the current user

MDCS_SECURITY_ManageUsersDlg

```
BOOL  MDCS_SECURITY_ManageUsersDlg(
HDBHANDLE hHandle,
HWND hWnd = NULL,
LPCSTR pszDlgTitle = NULL,
MDCS_ST_DlgUsersType eDlgType = MDCS_eDlgUsersSimple,
HICON hIcon = 0
);
```

Purpose

To call a dialog to manage users.

Parameters

hHandle - database connection handle
pszDlgTitle - dialog title
hWnd - handle to the application window
eDlgType - type of the dialog, if MDCS_eDlgUsersWithStorage will show storage assigned to each user
hIcon - dialog’s icon

Return

FALSE - if error occurred or the dialog was cancelled

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_SECURITY_ManageGroupsDlg

```
BOOL  MDCS_SECURITY_ManageGroupsDlg(
HDBHANDLE hHandle,
HWND hWnd,
LPCSTR pszDlgTitle,
BOOL bHideUserGroupsControl = TRUE,
BOOL* pbHideUserGroupsDefault = NULL,
HICON hIcon = 0
);
```

Purpose

Calls a dialog to manage groups.

Parameters

hHandle - database connection handle
pszDlgTitle - dialog title
hWnd - handle to application window
pbHideUserGroupsDefault - if true, will not show user-defined groups
bHideUserGroupsControl - if true, will hide check box for user-defined groups
hIcon - dialog's icon

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_SECURITY_GetUserGroups

```
BOOL  MDCS_SECURITY_GetUserGroups(
HDBHANDLE hHandle,
LONGLONG lUserID,
AxStringArray& arrGroupIDs,
AxStringArray& arrGroupNames
);
```

Purpose

Returns a list of groups that user has access to.

Parameters

hHandle - database connection handle
pResultCallback - pointer to a callback function to get data

Output

arrGroupIDs - group IDs
arrGroupNames - group names

Return

FALSE - if error occurred

MDCS_SECURITY_GetUserGroupsDlg

```
BOOL  MDCS_SECURITY_GetUserGroupsDlg(
HDBHANDLE hHandle,
HWND hWnd,
MDCS_GroupInfoCallback* pCallback
);
```

Purpose

Displays a selection dialog to choose groups.

Parameters

hHandle - database connection handle
pResultCallback - pointer to a callback function to group information
hWnd - handle to the application window

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_SECURITY_GetUserInfo

```
BOOL  MDCS_SECURITY_GetUserInfo(
HDBHANDLE hHandle,
LONGLONG lUserID,
MDCS_ST_UserInfo& stUserInfo);
```

Purpose

Returns information about a user.

Parameters

hHandle - database connection handle
lUserID - user ID, if 0, the function will return all info for current user

Output

stUserInfo - user information

Return

FALSE - if error occurred

MDCS_SECURITY_IsUserInAdminGroup

```
BOOL  MDCS_SECURITY_IsUserInAdminGroup(
HDBHANDLE hHandle,
LONGLONG lUserID,
BOOL& bAdmin
);
```

Purpose

To check if the user is in the Administrators group.

Parameters

hHandle - database connection handle
lUserID - user ID

Output

bAdmin - if TRUE, the user is in Administrators group

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_SECURITY_GetEveryoneGroupInfo

```
BOOL  MDCS_SECURITY_GetEveryoneGroupInfo (
HDBHANDLE hHandle,
MDCS_ST_GroupInfo& stGroupInfo);
);
```

Purpose

To get information about the Everyone group.

Parameters

hHandle - database connection handle

Output

stGroupInfo - group information

Return

FALSE - if error occurred

MDCS_SECURITY_ChangePassword

```
BOOL  MDCS_SECURITY_ChangePassword (
HDBHANDLE hHandle,
);
```

Purpose

To change the password, calls a dialog to allow the password change.

Parameters

hHandle - database connection handle

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Datasets Functions

12

This chapter contains functions that you can use to work with datasets.

Table 12-1: Dataset Functions

Function Name	Description
MDCS_DATASET_InsertAnalysisAttributes	To create/insert new dataset analysis attributes
MDCS_DATASET_GetAnalysisAttributesByDataset	To get all analysis attributes for a dataset
MDCS_DATASET_GetAnalysisAttributes	To get dataset analysis attributes per analysis
MDCS_DATASET_GetAnalysisDescriptions	To get all analysis descriptions per dataset
MDCS_DATASET_FindAnalysisAttributesRecord	To find a record by attributes
MDCS_DATASET_DeleteAnalysisAttributes	To delete analysis attributes
MDCS_DATASET_UpdateAnalysisAttributes	To update analysis attributes
MDCS_DATASET_ManageFolderSecurity	To call a dialog to manage security access to the dataset folders
MDCS_DATASET_DeleteFolder	To delete the dataset folder
MDCS_DATASET_Copy	To copy a dataset
MDCS_DATASET_Update	To update a dataset
MDCS_DATASET_Get	To get dataset information
MDCS_DATASET_Create	To create dataset
MDCS_DATASET_AddAssay	To add an assay to a dataset
MDCS_DATASET_HavePermissionsToModify	To check if the current user can modify a dataset
MDCS_DATASET_GetAnalysisDescription	To get dataset analysis configuration description.
MDCS_DATASET_GetAssayAndFilters	To get assays and filters that were used to create a dataset
MDCS_DATASET_GetAllInFolder	To get assays (with a basic description) in a folder
MDCS_DATASET_GetItemFolder	To get an item’s folder
MDCS_DATASET_GetAssaySiblingFolders	To get assay sibling folders
MDCS_DATASET_DoesSubFolderExist	To check if a subfolder exists in the database
MDCS_DATASET_CreateFolder	To create a new dataset folder in the database
MDCS_DATASET_ModifyFolder	To modify a dataset folder in the database
MDCS_DATASET_GetSiblingFolders	To get dataset sibling folders
MDCS_DATASET_GetAllMeasurements	To get data types for a dataset

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Table 12-1: Dataset Functions (cont'd)

MDCS_DATASET_AddScriptletAssay	To create and add a scriptlet to dataset
MDCS_DATASET_GetScriptletAssays	To get all scriptlets assays in dataset
MDCS_DATASET_GetScriptletAssayIDByName	To find a scriptlet by name in the dataset
MDCS_DATASET_GetAllForPlate	To get all available datasets for a plate
MDCS_DATASET_GetAllForAssay	To get all available datasets for an assay
MDCS_DATASET_GetAllMSetParamValues	To get the values of attributes in a dataset
MDCS_DATASET_GetResultsInfoByConfig	To get the dataset result info for a specified configuration
MDCS_DATASET_GetResultInfo	To get dataset result info
MDCS_DATASET_CallbackToAnalysisInfo	To convert the result of callback to MDCS_ST_DatasetResultInfo structure
MDCS_DATASET_DoesNameExist	To check if dataset with specified name already exists in the folder
MDCS_DATASET_UpdateFolderItem	To update the link to the folder of a dataset
MDCS_DATASET_Delete	To delete a dataset
MDCS_DATASET_GetAnalysisCount	To get a number of analysis per dataset

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_DATASET_InsertAnalysisAttributes

```
BOOL  MDCS_DATASET_InsertAnalysisAttributes(
HDBHANDLE hHandle,
const MDCS_ST_DsetAnalysis* pDsetAnalysis,
const MDCS_ST_DsetAnalysisAttr* pAttributes,
UINT uNumberAttr,
LONGLONG* plAnalysisID
);
```

Purpose

To create/insert new dataset analysis attributes.

Parameters

hHandle - database connection handle
pDsetAnalysis - description of analysis
pAttributes - pointer to the attributes array
uNumberAttr - number of attributes in array

Output

plAnalysisID - new analysis ID

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_DATASET_GetAnalysisAttributesByDataset

```
BOOL  MDCS_DATASET_GetAnalysisAttributesByDataset(
HDBHANDLE hHandle,
LONGLONG lDatasetID,
MDCS_GetDBResultsCCallback * pCallBack
);
```

Purpose

To get all analysis attributes for a dataset.

Parameters

hHandle - database connection handle
lDatasetID - dataset ID
pResultCallback - pointer to a callback function to get data

Output

Callback returns the following fields:
ATTR_ID - unique ID
CALC_METHOD - calculation method
ATTR_TYPE - attribute type (pivot, data type)
ANALYSIS_ID - analysis ID (pointer to a description of analysis)
ORDER_NUM - order number
COLLAPSE_COLUMN - column that is used to collapse data
ATTR_COLUMN - name of the attribute column

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_DATASET_GetAnalysisAttributes

```
BOOL  MDCS_DATASET_GetAnalysisAttributes(
HDBHANDLE hHandle,
LONGLONG lAnalysisID,
MDCS_GetDBResultsCCallback * pCallBack
);
```

Purpose

To get dataset analysis attributes per analysis.

Parameters

hHandle - database connection handle
lAnalsysisID - analysis ID
pResultCallback - pointer to a callback function to get data

Output

Callback returns the following fields:
ATTR_ID - unique ID
CALC_METHOD - calculation method
ATTR_TYPE - attribute type (pivot, data type)
ANALYSIS_ID - analysis ID (pointer to a description of analysis)
ORDER_NUM - order number
COLLAPSE_COLUMN - column that is used to collapse data
ATTR_COLUMN - name of the attribute column

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_DATASET_GetAnalysisDescriptions

```
BOOL  MDCS_DATASET_GetAnalysisDescriptions(
HDBHANDLE hHandle,
LONGLONG   lDatasetID,
MDCS_GetDBResultsCCallback * pCallBack
);
```

Purpose

To get all analysis descriptions per dataset.

Parameters

hHandle - database connection handle
lDatasetID - dataset ID
pResultCallback - pointer to a callback function to get data

Output

Callback returns the following fields:
ANALYSIS_ID - unique ID
DATASET_ID - ID of a dataset where it belongs
NAME - analysis description name
DESCRIPTION - description

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_DATASET_FindAnalysisAttributesRecord

```
BOOL  MDCS_DATASET_FindAnalysisAttributesRecord(
HDBHANDLE hHandle,
LONGLONG lDatasetID,
const MDCS_ST_DsetAnalysisAttr* pAttributes,
UINT uNumberAttr
MDCS_ST_DsetAnalysis* pDsetAttributes
);
```

Purpose

To find a record by attributes.

Parameters

hHandle - database connection handle
lDatasetID - dataset ID
pAttributes - pointer to the attributes array
uNumberAttr - number of attributes in array

Output

pDsetAttributes - record that corresponds to attributes

Return

FALSE - if error occurred

MDCS_DATASET_DeleteAnalysisAttributes

```
BOOL  MDCS_DATASET_DeleteAnalysisAttributes(
HDBHANDLE hHandle,
LONGLONG lAttributesID
);
```

Purpose

To delete analysis attributes.

Parameters

hHandle - database connection handle
lAttributesID - attributes ID

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_DATASET_UpdateAnalysisAttributes

```
BOOL  MDCS_DATASET_UpdateAnalysisAttributes(
HDBHANDLE hHandle,
const MDCS_ST_DsetAnalysis* pDsetAnalysis
);
```

Purpose

To update analysis attributes.

Parameters

hHandle - database connection handle
pDsetAnalysis - structure that contains attributes to update

Return

FALSE - if error occurred

MDCS_DATASET_ManageFolderSecurity

```
BOOL  MDCS_DATASET_ManageFolderSecurity(
HDBHANDLE hHandle,
LONGLONG lFolderID,
HWND hWnd = NULL,
LPCSTR pszDlgTitle = NULL
);
```

Purpose

Calls a dialog to manage security access to the dataset folders.

Parameters

hHandle - database connection handle
lFolderID - folder ID
pszDlgTitle - dialog title
hWnd - handle to the application window

Return

FALSE - if error occurred or dialog cancelled

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_DATASET_DeleteFolder

```
BOOL  MDCS_DATASET_DeleteFolder(
HDBHANDLE hHandle,
LONGLONG lFolderID
);
```

Purpose

Function to delete dataset folder.

Parameters

hHandle - database connection handle
lFolderID - ID of a folder in datasets tree

Return

FALSE - if error occurred

MDCS_DATASET_Copy

```
BOOL  MDCS_DATASET_Copy(
HDBHANDLE hHandle,
LONGLONG lDatasetSourceID,
LONGLONG lDestinationFolderID,
MDCS_ST_Dataset& stDataSetOut,
WORD wCopyDependants,
BOOL bCopySpots = TRUE,
BOOL bCopyAssays = TRUE
);
```

Purpose

Function to copy a dataset.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle

lDatasetSourceID - ID of the dataset that should be copied

lDestinationFolderID - ID of the folder where the dataset should be copied

wCopyDependants - option to copy dataset dependants such as results, objects, scriptlets (use MDC_E_DatasetObjectType)

bCopySpots - if TRUE, will copy all spots that were used in dataset

bCopyAssays - if TRUE, will copy all measurement sets associated with dataset

Output

stDataSetOut - dataset that was created

Return

FALSE - if error occurred

MDCS_DATASET_Update

```
BOOL  MDCS_DATASET_Update(
HDBHANDLE hHandle,
const MDCS_ST_Dataset& stDataSetIn
);
```

Purpose

Function to update a dataset.

Parameters

hHandle - database connection handle

stDataSetIn - dataset information that will be updated, should contain valid dataset ID

Return

FALSE - if error occurred

MDCS_DATASET_Get

```
BOOL  MDCS_DATASET_Get(
HDBHANDLE hHandle,
LONGLONG lDatasetID,
MDCS_ST_Dataset& stDataSetOut
);
```

Purpose

Function to get dataset information.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle
l*DatasetID* - dataset ID

Output

stDataSetOut - dataset information

Return

FALSE - if error occurred

MDCS_DATASET_Create

```
BOOL  MDCS_DATASET_Create(
HDBHANDLE hHandle,
LONGLONG lDestFolderID,
const MDCS_ST_Dataset& stDataSetIn,
MDCS_ST_Dataset& stDataSetOut
);
```

Purpose

Function to create a new dataset.

Parameters

hHandle - database connection handle
lDestFolderID - folder ID where dataset should be created
stDataSetIn - dataset description

Output

stDataSetOut - dataset information

Return

FALSE - if error occurred

MDCS_DATASET_AddAssay

```
BOOL  MDCS_DATASET_AddAssay(
HDBHANDLE hHandle,
LONGLONG lDatasetID,
LONGLONG lAssayID
);
```

Purpose

Function to create a new dataset assay.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle
lDatasetID - destination dataset ID
lAssayID - assay ID

Output

stDataSetOut - dataset information

Return

FALSE - if error occurred

MDCS_DATASET_HavePermissionsToModify

```
BOOL  MDCS_DATASET_HavePermissionsToModify(
HDBHANDLE hHandle,
LONGLONG lDatasetID,
BOOL& bCanModify
);
```

Purpose

To check if the current user can modify a dataset.

Parameters

hHandle - database connection handle
lDatasetID - dataset ID

Output

bCanModify - if TRUE, the user can modify the dataset

Return

FALSE - if error occurred

MDCS_DATASET_GetAnalysisDescription

```
BOOL  MDCS_DATASET_GetAnalysisDescription (
HDBHANDLE hHandle,
LONGLONG   lAnalyisisID,
MDCS_ST_DsetAnalysis& stDsetAttributes
);
```

Purpose

To get a dataset analysis configuration description.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle
lAnalyisisID - configuration ID

Output

stDsetAttributes - structure that describes the analysis configuration

Return

FALSE - if error occurred

MDCS_DATASET_GetAssayAndFilters

```
BOOL  MDCS_DATASET_GetAssayAndFilters (
HDBHANDLE hHandle,
LONGLONG lDatasetID,
AxStringArray& arrAssayIDs,
AxStringArray& arrAssayFilters
);
```

Purpose

To get assays and filters that were used to create a dataset.

Parameters

hHandle - database connection handle
lDatasetID - dataset ID

Output

arrAssayIDs - array of assay IDs
arrAssayFilters - array of assay filters

Return

FALSE - if error occurred

MDCS_DATASET_GetAllInFolder

```
BOOL  MDCS_DATASET_GetAllInFolder (
HDBHANDLE hHandle,
LONGLONG lFolderID,
MDCS_GetDBResultsCCallback* pResultCallback
);
```

Purpose

To get assays (with a basic description) in a folder.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle
lFolderID - folder ID
pResultCallback - pointer to a callback function to get data

Output

Fields that are always present:

DATASET_ID, DATASET_NAME, TIME_CREATED (as time_t value)
DATASET_DESCRIPTION - dataset description

DATASET_TYPE - type of the dataset

DATASET_DTYPE

ENGLISH_DESCR – dataset description

TO_DELETE

Return

FALSE - if error occurred

MDCS_DATASET_GetItemFolder

```
BOOL  MDCS_DATASET_GetItemFolder (
HDBHANDLE hHandle,
LONGLONG lItemID,
LONGLONG& lFolderID
);
```

Purpose

To get an item’s folder id.

Parameters

hHandle - database connection handle
lItemID - item ID

Output

lFolderID - folder ID

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_DATASET_GetAssaySiblingFolders

```
BOOL  MDCS_DATASET_GetAssaySiblingFolders (
HDBHANDLE hHandle,
LONGLONG lParentFolderID,
MDCS_GetDBResultsCCallback* pResultCallback
);
```

Purpose

Function to get assay sibling folders.

Parameters

hHandle - database connection handle
ParentFolderID - parent ID to retrieve siblings for
pResultCallback - pointer to a callback function to get data

Output

Result columns will be:
FOLDER_NAME - folder name
FOLDER_ID - folder ID

Return

FALSE - if error occurred

MDCS_DATASET_DoesSubFolderExist

```
BOOL  MDCS_DATASET_DoesSubFolderExist (
HDBHANDLE hHandle,
LONGLONG lFolderID,
LPCSTR pzName,
LONGLONG &lSubfolderID
);
```

Purpose

To check if the subfolder exists in the database.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle
lFolderID - parent folder ID to retrieve subfolder for
pzName - name of the parent folder

Output

lSubfolderID - subfolder ID, if 0 subfolder does not exists

Return

FALSE - if error occurred

MDCS_DATASET_CreateFolder

```
BOOL  MDCS_DATASET_CreateFolder (
HDBHANDLE hHandle,
const MDCS_ST_FolderInfo& stInfoIn,
MDCS_ST_FolderInfo & stInfoOut
);
```

Purpose

To create a new dataset folder in the database.

Parameters

hHandle - database connection handle
stInfoIn - structure describes the folder to be created in the database

Output

stInfoOut - structure that describes the new folder just created

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_DATASET_ModifyFolder

```
BOOL  MDCS_DATASET_ModifyFolder (
HDBHANDLE hHandle,
const MDCS_ST_FolderInfo& stInfoIn
);
```

Purpose

To modify a dataset folder in the database.

Parameters

hHandle - database connection handle
stInfoIn - structure that describes the folder to be modified

Output

NONE

Return

FALSE - if error occurred

MDCS_DATASET_GetSiblingFolders

```
BOOL  MDCS_DATASET_GetSiblingFolders (
HDBHANDLE hHandle,
LONGLONG lParentFolderID,
MDCS_GetDBResultsCCallback* pResultCallback
);
```

Purpose

To get dataset sibling folders.

Parameters

hHandle - database connection handle
lParentFolderID   - *parent* ID to retrieve siblings for

Output

pResultCallback - data callback
Result columns will be
FOLDER_NAME - folder name
FOLDER_ID - folder ID

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_DATASET_GetAllMeasurements

```
BOOL  MDCS_DATASET_GetAllMeasurements (
HDBHANDLE hHandle,
LONGLONG lDatasetID,
MDCS_GetDBResultsCCallback* pResultCallback
);
```

Purpose

To get data types for a dataset.

Parameters

hHandle - database connection handle
lDatasetID - dataset ID
pResultCallback - pointer to a callback function to get data

Output

Result columns will be:
TABLE_ID - table name
COLUMN_NAME
COLUMN_TYPE
COLUMN_NAME_EXT
FUNCTION_NAME
PARAMETER_NAME

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_DATASET_AddScriptletAssay

```
BOOL  MDCS_DATASET_AddScriptletAssay (
HDBHANDLE hHandle,
LONGLONG lDatasetID,
LONGLONG lScriptletID,
LPCSTR pszAssayName,
LONGLONG& lAssayIDOut
);
```

Purpose

To create and add a scriptlet to a dataset.

Parameters

hHandle - database connection handle
lDatasetID - dataset ID
lScriptletID - scriptlet assay ID
pszAssayName - name of assay

Output

pAssayIDOut   - created scriptlet assay ID

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_DATASET_GetScriptletAssays

```
BOOL  MDCS_DATASET_GetScriptletAssays (
HDBHANDLE hHandle,
LONGLONG lDatasetID,
LONGLONG lScriptletID,
AxStringArray& arrAssayIDs,
AxStringArray& arrAssayNames
);
```

Purpose

To get a scriptlet assay in a dataset.

Parameters

hHandle - database connection handle
lDatasetID - dataset ID
lScriptletID - assay scriptlet ID

Output

arrAssayIDs - array of scriptlet assay IDs
arrAssayNames - array of scriptlet assay names

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_DATASET_GetScriptletAssayIDByName

```
BOOL  MDCS_DATASET_GetScriptletAssayIDByName (
HDBHANDLE hHandle,
LONGLONG lDatasetID,
LPCSTR pszName,
LONGLONG& lAssayID
);
```

Purpose

To find a scriptlet assay ID by name.

Parameters

hHandle - database connection handle
lDatasetID - dataset ID
pszName - assay name

Output

lAssayID - assay ID

Return

FALSE - if error occurred

MDCS_DATASET_GetAllForPlate

```
BOOL  MDCS_DATASET_GetAllForPlate (
HDBHANDLE hHandle,
LONGLONG lPlateID,
MDCS_GetDBResultsCCallback * pResultCallback
);
```

Purpose

To get available datasets for a plate.

Parameters

hHandle - database connection handle
lPlateID - plate ID
pResultCallback - pointer to a callback function to get data

Output

Records include fields from the DATASET table.

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_DATASET_GetAllForAssay

```
BOOL  MDCS_DATASET_GetAllForAssay (
HDBHANDLE hHandle,
LONGLONG lAssayID,
MDCS_GetDBResultsCCallback * pResultCallback
);
```

Purpose

To get available datasets for an assay.

Parameters

hHandle - database connection handle
lAssayID - assay ID
pResultCallback - pointer to a callback function to get data

Output

Records include fields from the DATASET table.

Return

FALSE - if error occurred

MDCS_DATASET_GetAllMSetParamValues

```
BOOL  MDCS_DATASET_GetAllMSetParamValues (
HDBHANDLE hHandle,
LONGLONG lDatasetID,
const MDCS_ST_ScopeAttribute* arrAttributes,
INT_PTR nElemCount,
MDCS_GetDBResultsCCallback* pResultCallback
);
```

Purpose

To get values of attributes in a dataset.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

hHandle - database connection handle
lDatasetID - dataset ID
arrAttributes – list of attributes to retrieve values
nElemCount – number of attributes
pResultCallback - pointer to a callback function to get data

Output

Records include measurement set attribute values.

Return

FALSE - if error occurred

MDCS_DATASET_GetResultsInfoByConfig

```
BOOL  MDCS_DATASET_GetResultsInfoByConfig (
HDBHANDLE hHandle,
LONGLONG lDatasetID,
LONGLONG lConfigID,
MDCS_GetDBResultsCCallback * pCallBack
);
```

Purpose

To get description of results on dataset for specified configuration.

Parameters

hHandle - database connection handle
lDatasetID - dataset ID
lConfigID - configuration ID
pResultCallback - pointer to a callback function to get data

Output

Records include analysis info.

Return

FALSE - if error occurred

Note: Use MDCS_DATASET_CallbackToAnalysisInfo to process results.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_DATASET_GetResultInfo

```
BOOL  MDCS_DATASET_GetResultInfo (
HDBHANDLE hHandle,
LONGLONG lResultID,
MDCS_GetDBResultsCCallback * pCallBack
);
```

Purpose

To get dataset.results info.

Parameters

hHandle - database connection handle
lResultID - result ID
pResultCallback - pointer to a callback function to get data

Output

Records include analysis info

Return

FALSE - if error occurred

MDCS_DATASET_CallbackToAnalysisInfo

```
BOOL  MDCS_DATASET_CallbackToAnalysisInfo (
const MDCS_QueryResults* pQueryResults,
MDCS_ST_DatasetResultInfo& stOut,
BOOL bResultsInOrigForm = FALSE
);
```

Purpose

To convert the result of a callback to the *MDCS_ST_DatasetResultInfo* structure.

Parameters

pQueryResults - call back results
bResultsInOrigForm - if TRUE, the results are in the original form

Output

stOut - destination structure

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_DATASET_DoesNameExist

```
BOOL  MDCS_DATASET_DoesNameExist (
HDBHANDLE hHandle,
LPCSTR pszName,
LONGLONG lFolderID,
BOOL& bExists
);
```

Purpose

To check if the dataset exists.

Parameters

hHandle - database connection handle
pszName - dataset name
lFolderID - folder ID

Output

bExists - TRUE if exists

Return

FALSE - if error occurred

MDCS_DATASET_UpdateFolderItem

```
BOOL  MDCS_DATASET_UpdateFolderItem(
HDBHANDLE hHandle,
LONGLONG lFolderID,
LONGLONG lDatasetID,
);
```

Purpose

Function update the link to the folder of a dataset.

Parameters

hHandle - database connection handle
lDatasetID - dataset ID
lFolderID - folder ID

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_DATASET_Delete

```
BOOL  MDCS_DATASET_Delete(
HDBHANDLE hHandle,
LONGLONG lDatasetID,
);
```

Purpose

Function to delete a dataset.

Parameters

hHandle - database connection handle
lDatasetID - dataset ID

Return

FALSE - if error occurred

MDCS_DATASET_GetAnalysisCount

```
BOOL  MDCS_DATASET_GetAnalysisCount (
HDBHANDLE hHandle,
LONGLONG lDatasetID,
LONGLONG lCount,
);
```

Purpose

To get the number of analyses per dataset.

Parameters

hHandle - database connection handle
lDatasetID - dataset ID

Output

lCount - number of analyses

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Assay Images and Assay Normalization Functions

13

This chapter contains functions that you can use to work with assay images and assay normalization.

Table 13-1: Assay Images and Assay Normalization Functions

Function Name	Description
MDCS_ASSAYIMAGES_GetAllPropertyAttributes	To get the attributes that the measurement set images can be queried on
MDCS_ASSAYIMAGES_GetAllByAttributes	Function to get measurement set images based on specified attributes values
MDCS_ASSAYIMAGES_GetUniqueAttributeValues	To get unique values for a measurement set images attribute
MDCS_ASSAYNORM_CreateConfig	To create a new normalization configuration
MDCS_ASSAYNORM_UpdateConfig	To update an existing normalization configuration
MDCS_ASSAYNORM_GetConfig	To get a normalization configuration using its ID
MDCS_ASSAYNORM_GetConfigByName	To get a normalization configuration using its name
MDCS_ASSAYNORM_GetConfigAll	To get all normalization configurations
MDCS_ASSAYNORM_GetConfigForAssay	To get all normalization configurations that exist for an assay

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAYIMAGES_GetAllPropertyAttributes

```
BOOL  MDCS_ASSAYIMAGES_GetAllPropertyAttributes(
HDBHANDLE hHandle,
AxStringArray &arrayColumnIDs,
AxStringArray &arrayColumnNames
);
```

Purpose

To get the attributes that measurement set images can be queried on.

Parameters

hHandle - database connection handle
arrayColumnIDs - array to hold column ID
arrayColumnNames - array to hold column names

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAYIMAGES_GetAllByAttributes

```
BOOL  MDCS_ASSAYIMAGES_GetAllByAttributes(
HDBHANDLE hHandle,
const AxStringArray& arrayAssayIDs,
const AxStringArray* pAXDisplayColumns,
const AxStringArray* pAXAttrColumn,
const AxStringArray* pAXAttrValues,
MDCS_GetDBResultsCCallback* pResultCallback
);
```

Purpose

Function to get measurement set images based on specified attributes values.

Parameters

hHandle - database connection handle
arrayAssayIDS - assay IDs
pAXDisplayColumns - columns to retrieve
pAXAttrColumn - columns to query on
pAXAttrValues - values of the column to query on
pResultCallback - pointer to a callback function to get data

Output

Result columns are:
OBJ_ID - object ID
PLATE_ID - plate ID
IMAGE_ID - plate ID
SITE_ID - site ID
SERIES_ID - series ID
WELL_X - well X
WELL_Y - well Y
X_POSITION - x position
Y_POSITION - y position
Z_INDEX - z-index
Z_POSITION - z- position
T_INDEX
T_POSITION - T position in seconds
T_MICROSECONDS

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAYIMAGES_GetUniqueAttributeValues

```
BOOL  MDCS_ASSAYIMAGES_GetUniqueAttributeValues(
HDBHANDLE hHandle,
const AxStringArray& arrayAssayIDs,
LPCSTR pszAttrColumnName,
const AxStringArray* pAXAttrColumnNames,
const AxStringArray* pAXAttrValues,
AxStringArray* pAXValuesOut
);
```

Purpose

Function to get unique values for a measurement set images attribute.

Parameters

hHandle - database connection handle
arrayAssayIDs - array of assay IDs
pszAttrColumnName - column that will contain unique values
pAXAttrColumnNames - attribute column names to query on
pAXAttrValues - attribute values

Output

pAXValuesOut - array of unique values

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAYNORM_CreateConfig

```
BOOL  MDCS_ASSAYNORM_CreateConfig (
HDBHANDLE hHandle,
const MDCS_ST_NormConfig& stConfigIn,
MDCS_ST_NormConfig* pstConfigOut
);
```

Purpose

Function to a create new normalization configuration.

Parameters

hHandle - database connection handle
stConfigIn - configuration to create

Output

pstConfigOut - created configuration

Return

FALSE - if error occurred

MDCS_ASSAYNORM_UpdateConfig

```
BOOL  MDCS_ASSAYNORM_UpdateConfig (
HDBHANDLE hHandle,
const MDCS_ST_NormConfig& stConfigIn
);
```

Purpose

Function to update a normalization configuration.

Parameters

hHandle - database connection handle
stConfigIn - configuration that will be updated (ID must exist)

Output

NONE

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAYNORM_GetConfig

```
BOOL  MDCS_ASSAYNORM_GetConfig (
HDBHANDLE hHandle,
LONGLONG lConfigID,
MDCS_ST_NormConfig& stConfigOut
);
```

Purpose

Function to get a normalization configuration using its ID.

Parameters

hHandle - database connection handle
lConfigID - configuration ID

Output

stConfigOut - configuration for *lConfigID*

Return

FALSE - if error occurred

MDCS_ASSAYNORM_GetConfigByName

```
BOOL  MDCS_ASSAYNORM_GetConfigByName (
HDBHANDLE hHandle,
LPCSTR pszConfigName,
MDCS_ST_NormConfig& stConfigOut
);
```

Purpose

Function to get a normalization configuration using its name.

Parameters

hHandle - database connection handle
pszConfigName - Name of configuration to retrieve

Output

stConfigOut - configuration for pszConfigName

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_ASSAYNORM_GetConfigAll

```
BOOL  MDCS_ASSAYNORM_GetConfigAll (
HDBHANDLE hHandle,
MDCS_GetDBResultsCCallback* pResultCallback
);
```

Purpose

Function to get all existing normalization configurations.

Parameters

hHandle - database connection handle
pResultCallback - pointer to a callback function to get data

Output

Returns columns that could be mapped to the *MDCS_ST_NormConfig* structure.

Return

FALSE - if error occurred

MDCS_ASSAYNORM_GetConfigForAssay

```
BOOL  MDCS_ASSAYNORM_GetConfigForAssay (
HDBHANDLE hHandle,
LONGLONG lAssayID,
MDCS_GetDBResultsCCallback* pResultCallback
);
```

Purpose

Function to get all existing normalization configurations for an assay.

Parameters

hHandle - database connection handle
lAssayID - assay ID
pResultCallback - pointer to a callback function to get data

Output

Return columns that could be mapped to the *MDCS_ST_NormConfig* structure.

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Quicklist Functions

14

This chapter contains functions that you can use to work with quicklists.

Table 14-1: Quicklist Functions

Function Name	Purpose
MDCS_QUICKLIST_ManageFolderSecurity	Call a dialog to manage security access to the quicklist folders

MDCS_QUICKLIST_ManageFolderSecurity

```
BOOL  MDCS_QUICKLIST_ManageFolderSecurity(
HDBHANDLE hHandle,
LONGLONG lFolderID,
HWND hWnd = NULL,
LPCSTR pszDlgTitle = NULL
);
```

Purpose

Calls a dialog to manage security access to the quicklist folders.

Parameters

hHandle - database connection handle
lFolderID - folder ID
pszTitle - dialog title
hWnd - handle to the application window

Return

FALSE - if dialog was cancelled

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Application Data Table Functions

15

This chapter contains functions that you can use to work with application data tables.

Table 15-1: Application Data Table Functions

Function Name	Description
MDCS_APP_ManageLocationOptions	To call a dialog that allows the user to edit location options
MDCS_APP_CreateLocationOption	To create a new record that describes the location option
MDCS_APP_CreateJobRecord	To create a new job
MDCS_APP_CreateParameter	To create a new application parameter
MDCS_APP_GetParameterByName	To get free space in a file location
MDCS_APP_GetParameterByName	To get an application parameter using its name
MDCS_APP_GetParameterByID	To get an application parameter using its ID
MDCS_APP_UpdateParameter	To update an application parameter using its ID
MDCS_APP_GetLocationOptions	To get all records from the LOCATION_OPTIONS table
MDCS_APP_GetLocationOptionRecord	To get a single record from the LOCATION_OPTIONS table
MDCS_APP_GetLocationRecordsByLabel	To get all records from location table filtered by Label value
MDCS_APP_DeleteLocationOption	To delete a location option by its ID
MDCS_APP_DeleteUserLocationOptions	To delete a user location options by the user login
MDCS_APP_SetUserLocationOptions	To assign a user location option
MDCS_APP_GetUserLocationOption	To get a default location option
MDCS_APP_GetJobQueueRecords	To get all records from the JOB_QUEUE table
MDCS_APP_GetJobQueueRecord	To get a record from the JOB_QUEUE table.
MDCS_APP_GetJobQueue	To get job queue information using the JOB ID
MDCS_APP_UpdateJobStatus	To update job status
MDCS_APP_UpdateJobProgress	To update job progress
MDCS_APP_UpdateJobQueueRecord	To update job queue information using the job ID
MDCS_APP_CancelJobProgress	To cancel job
MDCS_APP_ClaimJob	To claim a new job
MDCS_APP_ResetJob	To reset job status and progress

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Table 15-1: Application Data Table Functions (cont'd)

MDCS_APP_ResetCrashedJobs	To reset job status and progress of all crashed jobs from last run
MDCS_APP_RefreshAllJobs	To refresh the status of all jobs
MDCS_APP_GetLocationOptionByID	To get a single record from the LOCATION_OPTIONS table by ID

MDCS_APP_ManageLocationOptions

```
BOOL  MDCS_APP_ManageLocationOptions(
HDBHANDLE hHandle,
MDCS_ST_LocationOption* pstLocationOption,
HWND hWnd = NULL,
const LONGLONG* plDefaultSelectionID = NULL,
LPCSTR pszDlgTitle = NULL,
HICON hIcon = 0
BOOL bSelectionMode = TRUE
);
```

Purpose

Function calls dialog that allows user editing of location options.

Parameters

hHandle - database connection handle

hWnd - handle to the parent window

plDefaultSelectionID - optional ID for the record that should be selected by default in dialog

pszDlgTitle - dialog title

hIcon - dialog's icon

bSelectionMode - if TRUE, dialog is called in the selection mode (otherwise just display mode)

Output

pstLocationOption - selected location option

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_APP_CreateLocationOption

```
BOOL MDCS_APP_CreateLocationOption(
HDBHANDLE hHandle,
const MDCS_ST_LocationOption& stLocation,
LONGLONG* plLocationID
);
```

Purpose

Function to create a new record that describes a location option.

Parameters

hHandle - database connection handle
stLocation - location description

Output

plLocationID - location ID for the new location option

Return

FALSE - if error occurred

MDCS_APP_CreateJobRecord

```
BOOL MDCS_APP_CreateJobRecord(
HDBHANDLE hHandle,
const MDCS_ST_Job& stJob,
LONGLONG* plJobID
);
```

Purpose

Function to create a new job.

Parameters

hHandle - database connection handle
stJob - job description

Output

plJobID - Job ID for new job record

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_APP_CreateParameter

```
BOOL  MDCS_APP_CreateParameter(
HDBHANDLE hHandle,
const MDCS_ST_AppParameter& stParam,
LONGLONG* plParamID
);
```

Purpose

Function to create a new application parameter.

Parameters

hHandle - database connection handle
stParam - application parameter

Output

plParamID - Parameter ID for new parameter

Return

FALSE - if error occurred

MDCS_APP_GetParameterByName

```
BOOL  MDCS_APP_GetFreeSpaceFromFileLocation(
HDBHANDLE hHandle,
const MDCS_ST_LocationOption& stLocation,
LONGLONG* lFreeSpace
);
```

Purpose

Function to get free disk space in a file location.

Parameters

hHandle - database connection handle
stLocation - location description

Output

lFreeSpace - free disk space (in kilobytes)

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_APP_GetParameterByName

```
BOOL  MDCS_APP_GetParameterByName(
HDBHANDLE hHandle,
LPCSTR pszParamName,
MDCS_ST_AppParameter& stParam
);
```

Purpose

Function to get an application parameters using its name.

Parameters

hHandle - database connection handle
pszParamName - application parameter name

Output

stParam - application parameter information structure

Return

FALSE - if error occurred

MDCS_APP_GetParameterByID

```
BOOL  MDCS_APP_GetParameterByID(
HDBHANDLE hHandle,
LONGLONG lParamID,
MDCS_ST_AppParameter& stParam
);
```

Purpose

Function to get application parameter using its ID.

Parameters

hHandle - database connection handle
lParamID - application parameter ID

Output

stParam - application parameter information structure

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_APP_UpdateParameter

```
BOOL  MDCS_APP_UpdateParameter(
HDBHANDLE hHandle,
LONGLONG lParamID,
const MDCS_ST_AppParameter& stParam
);
```

Purpose

Function to update application parameter using its ID.

Parameters

hHandle - database connection handle
lParamID - application parameter ID
stParam - application parameter information structure

Return

FALSE - if error occurred.

MDCS_APP_GetLocationOptions

```
BOOL  MDCS_APP_GetLocationOptions(
HDBHANDLE hHandle,
MDCS_GetDBResultsCCallback* pResultCallback
);
```

Purpose

Function to get all records from the LOCATION_OPTIONS table.

Parameters

hHandle - database connection handle
pResultCallback - callback object

Output

Columns from LOCATION_OPTIONS table

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_APP_GetLocationOptionRecord

```
BOOL  MDCS_APP_GetLocationOptionRecord(
HDBHANDLE hHandle,
LONGLONG lLocationID,
MDCS_GetDBResultsCCallback* pResultCallback
);
```

Purpose

Function to get a single record from the LOCATION_OPTIONS table.

Parameters

hHandle - database connection handle
pResultCallback - pointer to a callback function to get data
lLocationID - location option ID

Output

Columns from LOCATION_OPTIONS table

Return

FALSE - if error occurred

MDCS_APP_GetLocationRecordsByLabel

```
BOOL  MDCS_APP_GetLocationRecordsByLabel(
HDBHANDLE hHandle,
LPCSTR pszLabel,
MDCS_GetDBResultsCCallback* pResultCallback
);
```

Purpose

Function to get all records from the LOCATION_OPTIONS table filtered by Label value.

Parameters

hHandle - database connection handle
pszLabel - label
pResultCallback - pointer to a callback function to get data

Output

Columns from LOCATION_OPTIONS table

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_APP_DeleteLocationOption

```
BOOL  MDCS_APP_DeleteLocationOption(
HDBHANDLE hHandle,
LONGLONG lOptionID
);
```

Purpose

Function to delete a location option by its ID.

Parameters

hHandle - database connection handle
lOptionID - location option ID

Return

FALSE - if error occurred

MDCS_APP_DeleteUserLocationOptions

```
BOOL  MDCS_APP_DeleteUserLocationOptions(
HDBHANDLE hHandle,
LONGLONG lUserID
);
```

Purpose

Function to delete user location options by user login.

Parameters

hHandle - database connection handle
lUserID - user ID

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_APP_SetUserLocationOptions

```
BOOL  MDCS_APP_SetUserLocationOptions(
HDBHANDLE hHandle,
LONGLONG lOptionID,
LONGLONG lUserID,
MDCS_E_UserDataLocationOptions eType =
MDCS_eUserDataOptionDefault
);
```

Purpose

Function to assign user location options.

Parameters

hHandle - database connection handle
lOptionID - option ID
lUserID - user ID
eType - type of the option (MDCS_eUserDataOptionDefault - by default)

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_APP_GetUserLocationOption

```
BOOL  MDCS_APP_GetUserLocationOption(
HDBHANDLE hHandle,
LONGLONG lUserID,
MDCS_ST_LocationOption &stLocationOption,
MDCS_E_UserDataLocationOptions eType =
MDCS_eUserDataOptionDefault
);
```

Purpose

Function to get the default location option.

Parameters

hHandle - database connection handle
lOptionID - option ID
lUserID - user ID
eType - type of the option (MDCS_eUserDataOptionDefault - by default).

Output

stLocationOption - location option

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_APP_GetJobQueueRecords

```
BOOL  MDCS_APP_GetJobQueueRecords(
HDBHANDLE hHandle,
MDCS_GetDBResultsCCallback* pResultCallback
);
```

Purpose

To get all records from the JOB_QUEUE table.

Parameters

hHandle - database handle
pResultCallback - pointer to a callback function to get data

Output

Returns columns:
ID
TIMESTAMP - as time updated
All other columns from JOB_QUEUE table.

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_APP_GetJobQueueRecord

```
BOOL  MDCS_APP_GetJobQueueRecord(
HDBHANDLE hHandle,
LONGLONG lJobID,
MDCS_GetDBResultsCCallback* pResultCallback
);
```

Purpose

To get all records from the JOB_QUEUEtable.

Parameters

hHandle - database handle
lJobID - job ID
pResultCallback - pointer to a callback function to get data

Output

NULL columns:
ID
TIMESTAMP - as time updated
All other columns from JOB_QUEUE table.

Return

FALSE - if error occurred

MDCS_APP_GetJobQueue

```
BOOL  MDCS_APP_GetJobQueue(
HDBHANDLE hHandle,
LONGLONG lJobID,
MDCS_ST_Job& stJob
);
```

Purpose

To get information from the JOB_QUEUE table using the job ID.

Parameters

hHandle - database connection handle
lJobID - job ID

Output

stJob - structure that contains job information

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_APP_UpdateJobStatus

```
BOOL  MDCS_APP_UpdateJobStatus(
HDBHANDLE hHandle,
LONGLONG lJobID,
LPCSTR pszStatus
);
```

Purpose

To update job status.

Parameters

hHandle - database connection handle
lJobID - job ID
pszStatus - status

Output

NONE

Return

FALSE - if error occurred

MDCS_APP_UpdateJobProgress

```
BOOL  MDCS_APP_UpdateJobProgress(
HDBHANDLE hHandle,
LONGLONG lJobID,
LPCSTR pszProgress
);
```

Purpose

To update job progress.

Parameters

hHandle - database connection handle
lJobID - job ID
pszProgress - Progress

Output

NONE

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_APP_UpdateJobQueueRecord

```
BOOL  MDCS_APP_UpdateJobQueueRecord(
HDBHANDLE hHandle,
LONGLONG lJobID,
const MDCS_ST_Job& stUpdateJob
);
```

Purpose

To update job queue information using the job ID.

Parameters

hHandle - database connection handle
lJobID - job ID
stUpdateJob - information to be updated

Output

NONE

Return

FALSE - if error occurred

MDCS_APP_CancelJobProgress

```
BOOL  MDCS_APP_CancelJobProgress(
HDBHANDLE hHandle,
LONGLONG lJobID
);
```

Purpose

To cancel job.

Parameters

hHandle - database connection handle
lJobID - job ID

Output

NONE

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_APP_ClaimJob

```
BOOL  MDCS_APP_ClaimJob(
HDBHANDLE hHandle,
LONGLONG lJobID
BOOL dDistributed = FALSE
);
```

Purpose

To claim a new job.

Parameters

hHandle - database connection handle
dDistributed - if TRUE, analysis environment is distributed

Output

lJobID - job ID

Return

FALSE - if error occurred

MDCS_APP_ResetJob

```
BOOL  MDCS_APP_ResetJob(
HDBHANDLE hHandle,
LONGLONG lJobID
);
```

Purpose

To reset job status and progress.

Parameters

hHandle - database connection handle
lJobID - job ID

Output

NONE

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_APP_ResetCrashedJobs

```
BOOL  MDCS_APP_ResetJobProgress(
HDBHANDLE hHandle,
LPCSTR pszNetWorkID
);
```

Purpose

To reset job status and progress of all the crashed jobs from the last run.

Parameters

hHandle - database connection handle
pszNetworkID - network ID

Output

NONE

MDCS_APP_RefreshAllJobs

```
BOOL  MDCS_APP_RefreshAllJobs(
HDBHANDLE hHandle
);
```

Purpose

To refresh the status of all jobs.

Parameters

hHandle - database connection handle

Output

NONE

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_APP_GetLocationOptionByID

```
BOOL  MDCS_APP_GetLocationOptionByID (
HDBHANDLE hHandle,
LONGLONG lLocationID,
MDCS_ST_LocationOption& stLocationOption
);
```

Purpose

To get a single record from LOCATION_OPTIONS table using the ID.

Parameters

hHandle - database connection handle
lLocationID - location option ID

Output

stLocationOption - location option structure

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

File Data and Macro Functions

16

This chapter contains functions that you can use to work with file data and macros.

Table 16-1: File Data and Macro Functions

Function Name	Description
MDCS_FILEDATA_GetRecordByID	To find record ID by GUID
MDCS_FILEDATA_GetRecordByGUID	To find file data record ID by GUID
MDCS_FILEDATA_UpdateRecord	To update a file data record
MDCS_MACROS_GetRecord	To get macros information by macros ID
MDCS_MACROS_GetAllByGroup	To get all macros by group
MDCS_MACROS_UpdateRecord	To update a macros record
MDCS_MACROS_DeleteRecord	To delete a macros record
MDCS_MACROS_Create	To create a new macros
MDCS_MACROS_ConvertQueryResultsToMacrosStructure	To convert the results of a query callback to a macros structure

MDCS_FILEDATA_GetRecordByID

```
BOOL  MDCS_FILEDATA_GetRecordByID(
HDBHANDLE hHandle,
LONGLONG lID,
MDCS_ST_FileDataInfo* pstFileData
);
```

Purpose

Function to find the record ID by GUID.

Parameters

hHandle - database connection handle
lID - record ID

Output

pstFileData - file data

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_FILEDATA_GetRecordByGUID

```
BOOL  MDCS_FILEDATA_GetRecordByGUID(
HDBHANDLE hHandle,
LPCSTR pszGUID,
MDCS_ST_FileDataInfo* pstFileData
);
```

Purpose

Function to find the file data record ID by GUID.

Parameters

hHandle - database connection handle
pszGUID - GUID

Output

pstFileData - file data

Return

FALSE - if error occurred

MDCS_FILEDATA_UpdateRecord

```
BOOL  MDCS_FILEDATA_UpdateRecord(
HDBHANDLE hHandle,
LONGLONG lID,
const MDCS_ST_FileDataInfo& stFileData
);
```

Purpose

Function to update a file data record.

Parameters

hHandle - database connection handle
lID - record ID
pstFileData - file data

Output

NONE

Return

FALSE – if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_MACROS_GetRecord

```
BOOL  MDCS_MACROS_GetRecord (
HDBHANDLE hHandle,
LONGLONG lMacrosID,
MDCS_ST_GroupMacros& stMacrosData
);
```

Purpose

Function to get macros information by macros ID.

Parameters

hHandle - database connection handle
lMacrosID - macros ID

Output

stMacrosData - structure describes macros data

Return

FALSE – if error occurred

MDCS_MACROS_GetAllByGroup

```
BOOL  MDCS_MACROS_GetAllByGroup (
HDBHANDLE hHandle,
LONGLONG lGroupID,
MDCS_GetDBResultsCCallback* pResultCallback
);
```

Purpose

Function to get all macros by group.

Parameters

hHandle - database connection handle
lGroupID - Group ID

Output

pResultCallback - callback with data of all macros

Return

FALSE – if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_MACROS_UpdateRecord

```
BOOL  MDCS_MACROS_UpdateRecord (
HDBHANDLE hHandle,
LONGLONG lMacrosID,
const MDCS_ST_GroupMacros& stMacrosData
);
```

Purpose

Function to update the macros record.

Parameters

hHandle - database connection handle
lMacrosID - macros ID
stMacrosData - updated macros data
pResultCallback - pointer to a callback function to get data

Output

All macro data

Return

FALSE – if error occurred

MDCS_MACROS_DeleteRecord

```
BOOL  MDCS_MACROS_DeleteRecord (
HDBHANDLE hHandle,
LONGLONG lMacrosID,
);
```

Purpose

Function to delete the macros record.

Parameters

hHandle - database connection handle
lMacrosID - macros ID

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

MDCS_MACROS_Create

```
BOOL  MDCS_MACROS_Create(
HDBHANDLE hHandle,
const MDCS_ST_GroupMacros& stObjIn,
MDCS_ST_GroupMacros& stObjOut
);
```

Purpose

Function to create a new macros.

Parameters

hHandle - database connection handle
stObjIn - macros data to insert

Output

stObjOut - created macros

Return

FALSE - if error occurred

MDCS_MACROS_ConvertQueryResultsToMacrosStructure

```
BOOL  MDCS_MACROS_ConvertQueryResultsToMacrosStructure (
MDCS_QueryResults* pResults,
MDCS_ST_GroupMacros& stMacros,
BOOL bInNativeFormat = FALSE
);
```

Purpose

Function to convert results of query callback to macros structure.

Parameters

pResults - pointer to query results object
bInNativeFormat - if TRUE, the query results should be treated in their original format otherwise they will be treated as strings

Output

stMacros - converted data

Return

FALSE - if error occurred

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Structures, ENUMS, and Definitions 17

This chapter contains the following topics:

Structures

The MDCStoreUtils API uses many user-defined data types whose names and descriptions are provided in Table 17-1.

Table 17-1: Structures

Name	Description
1. MDCS_ST_COLUMNPROP	Structure of the column from TABLE_COLUMNS table (represents MDCStore data type information)
2. MDCS_ST_BlobLocation	Describes BLOB location on server
3. MDCS_ST_BlobInfo	Describes basic BLOB information
4. MDCS_ST_ShapeInfo	Describes measurement set result value
5. MDCS_ST_UserLogin	Describes user credentials to connect to the database
6. MDCS_ST_UserInfo	Describes a user
7. MDCS_ST_GroupInfo	Describes a group
8. MDCS_ST_FolderGroups	Describes group access to the folder
9. MDCS_ST_ObjGroups	Describes group access to a plate
10. MDCS_ST_Result set	Common structure to describe objects (result set) in the database
11. MDCS_ST_DBVersion	Describes the database version
12. MDCS_ST_Measurements	Describes measurements
13. MDCS_ST_Assay	Describes measurement sets
14. MDCS_ST_PlateImages	Describes a plate image
15. MDCS_ST_ShapeLine	Describes shape lines
16. MDCS_ST_ShapeLinesBlob	Describes the BLOB that contains shape lines
17. MDCS_ST_DsetAnalysis	Describes dataset analysis
18. MDCS_ST_DsetAnalysisAttr	Describes dataset analysis attributes
19. MDCS_ST_FolderInfo	Describes a folder
20. MDCS_ST_PlateInfo	Describes a plate
21. MDCS_ST_SiteImageInfo	Describes the site images
22. MDCS_ST_SeriesInfo	Describes series information

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Table 17-1: Structures (cont’d)

23. MDCS_ST_Dataset	Describes dataset information
24. MDCS_ST_Site	Describes the site
25. MDCS_ST_ImageSource	Describes the image source
26. MDCS_ST_Acquisition	Describes an acquisition
27. MDCS_ST_AcquisitionProfile	Describes an acquisition profile
28. MDCS_ST_AssayRun	Describes an assay runs
29. MDCS_ST_AssayProfile	Describes an assay profile
30. MDCS_ST_LocationOption	Describes a location option
31. MDCS_ST_Job	Describes a job
32. MDCS_ST_AppParameter	Describes an application parameter
33. MDCS_ST_PlateTemplate	Describes a plate layout template
34. MDCS_ST_ConnectionInfo	Describes database connection information
35. MDCS_ST_Attribute	Describes attributes of the plate, dataset, measurement set
36. MDCS_ST_FileDataInfo	Describes the file data records
37. MDCS_ST_WellInfo	Describes well information
38. MDCS_ST_CellDescription	Describes a measurement
39. MDCS_ST_BlobInfoEx	Describes extended BLOB info
40. MDCS_ST_ScopeAttribute	Describes scope attribute of a plate, dataset or measurement set
41. MDCS_ST_GroupMacros	Describes group macros
42. MDCS_ST_Warehouse	Describes annotation warehouse
43. MDCS_ST_NormConfig	Describes normalization configuration
44. MDCS_ST_MeasurementUseStatistic	Describes measurement user statistic
45. MDCS_ST_DatasetResultInfo	Describes dataset analysis results
46. MDCS_ST_DlgParams	Describes the parameters needed in a dialog

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

ENUMS Types

Table 17-2: ENUM Types

Name	Description
1. MDCS_E_SQLServerType	Describes the type of database server
2. MDCS_E_DBObjectType	Describes the type of object in the database
3. MDCS_E_FileStorage	Describes the location where files should be stored
4. MDCS_E_RatioType	Describes ratio types
5. MDCS_E_Waves	Describes waves
6. MDCS_E_NormalizationType	Describes normalization types
7. MDCS_E_AverageType	Describes data average types
8. MDCS_E_ColumnType	Describes column types
9. MDCS_E_ImageType	Describes image types
10. MDCS_E_BlobType	Describes BLOB types
11. MDCS_E_ShapeType	Describes shapes
12. MDCS_E_AttributeType	Describes attribute types
13. MDSC_E_StatisticType	Describes statistical types
14. MDSC_E_CollapseType	Describes collapse types
15. MDCS_E_UserStatus	Describes user status to access data
16. MDSC_E_GroupType	Describes group types
17. MDCS_E_AccessPermissions	Describes access permissions to data
18. MDCS_E_PlateAccessPermissions	Describes access permissions to plates
19. MDCS_E_CompareOperation	Describes available comparison operations
20. MDCS_E_DatasetType	Describes dataset types
21. MDCS_E_Statistic	Describes defined statistical methods
22. MDCS_E_UserDataLocationOptions	Describes user data location options
23. MDCS_ST_DlgUsersType	Describes the dialog types to manage users
24. MDC_E_DatasetDependantType	Describes the dataset dependent type
25. MDC_E_SQLServerVersion	Describes version of the database server
26. MDCS_E_GroupObjects	Describes group objects
27. MDCS_E_GroupObjectsType	Describes group object types
28. MDCS_E_DataType	Describes type of result data
29. MDCS_E_DSFilteringGroup	Describes types of group objects that can be used in datasets
30. MDCS_E_MeasurementGroupBy	Describes measurement group by variations
31. MDCS_E_PlateElements	Describes measurement group by variations
32. MDCS_E_NormConfigType	Describes type of normalization configuration

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Table 17-2: ENUM Types (cont'd)

33. MDCS_QueryBldType	Describes type of query builder

Definitions

Table 17-3: Max Object Size Definitions

Name	Value	Description
1. MDCS_MAX_DATABASE_ID	20	Max size of the unique database ID
2. MDCS_MAX_OBJ_ID	20	Max size of the object ID
3. MDCS_MAX_COL_NAME	20	Max size of the database column name (user defined, i.e. data types)
4. MDCS_MAX_COL_NAMEEXT	128	Max size of the ext field
5. MDCS_MAX_COL_RATIOTYPE	4	Max size of the ratio type
6. MDCS_MAX_COL_WAVEDEF	4	Max type of the wavelength field
7. MDCS_MAX_COL_NORMTYPE	4	Max size of the normalization type field
8. MDCS_MAX_TABLEID	25	Max size of the table name in the database
9. MDCS_MAX_RESULT SETID	18	Max size of the result set ID
10. MDCS_MAX_COL_DESCRCOLUMN	2000	Max size of the description field
11. MDCS_MAX_COL_ORDERNUM	16	Max size of the order num field
12. MDCS_MAX_COL_TYPE	10	Max size of the type field
13. MDCS_MAX_COL_LENGTH	4	Max length of the column in database (numeric)
14. MDCS_MAX_COL_AVGTYPE	4	Max size of the column that contains average type description
15. MDCS_MAX_COL_NORMFACTOR	50	Max size of the column that contains normalization factor
16. MDCS_MAX_COL_DEFAULT	10	Max size of the column that contains defaults
17. MDCS_MAX_DSN	256	Max size of the DSN (ODBC data source name)
18. MDCS_MAX_USERNAME	50	Max size of the user name
19. MDCS_MAX_PASSW	256	Max size of the user password
20. MDCS_MAX_DESCRIPTION	256	Max size of the description field
21. MDCS_MAX_OBJ_NAME	128	Max size of the object name
22. MDCS_MAX_TABLE_NAME	25	Max size of the table name
23. MDCS_MAX_SERVER	256	Max size of the server name
24. MDCS_MAX_DBNAME	256	Max size of the database name
25. MDCS_MAX_ERR_SIZE	1024	Max size of error message
26. MDCS_MAX_CONNECTION_STRING	1000	Max size of connection string
27. MDCS_MAX_BARCODE	256	Max size of a barcode

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Table 17-3: Max Object Size Definitions (cont'd)

28. MDCS_MAX_GLOBAL_ID	256	Max size of a global ID
29. MDCS_MAX_WELL_NAME	25	Max size of a well name
30. MDCS_MAX_UNIT_NAME	25	Max size of a unit name
31. MDCS_MAX_COMPOUND_NAME	128	Max size of a compound name
32. MDCS_MAX_CELL_ID	128	Max size of a cell ID
33. MDCS_MAX_DESCRIPTION_LARGE	1024	Max size of a large description
34. MDCS_MAX_STORAGE_TYPE	256	Max size of a storage type
35. MDCS_MAX_LABEL	256	Max size of a label
36. MDCS_MAX_NETWORK_ID	256	Max size of network ID
37. MDCS_MAX_REQUEST	20	Max size of request
38. MDCS_MAX_PARAM_VALUE	255	Max size of parameter value
39. MDCS_MAX_SERVER_VERSION	255	Max size of database server version
40. MDCS_MAX_SERVER_REVISION	255	Max size of database server revision, i.e. service pack
41. MDCS_MAX_SERVER_TYPE	255	Max size of database server type
42. MDCS_MAX_ODBC_DRIVER_NAME	255	Max size of ODBC driver name
43. MDCS_MAX_ODBC_DRIVER_VERSION	255	Max size of ODBC driver version
44. MDCS_MAX_ODBC_VERSION	255	Max size of ODBC version
45. MDCS_MAX_ODBC_SOURCE_NAME	255	Max size of ODBC datasource name
46. MDCS_MAX_GUID	256	Max size of GUID
47. MDCS_MAX_OBJ_NAME_BIG	255	Max size of the object name

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Error Code Definitions

Table 17-4: Error Code Definitions

Name	Value	Description
1. MDCS_ERR_CANCEL	2999	Operation was cancelled
2. MDCS_ERR_SUCCESS	3000	Operation completed successfully without error
3. MDCS_ERR_ODBC	3001	Error was reported by ODBC driver
4. MDCS_ERR_DATABASE_INCORRECT_PARAM	3002	Incorrect parameter was supplied to function
5. MDCS_ERR_DATABASE_OBJECT_NOT_EXIST	3003	Database object does not exist
6. MDCS_ERR_USER_PERMISSIONS	3004	User does not have sufficient privileges to access data, permission denied
7. MDCS_ERR_INCORRECT_OBJ_SIZE	3005	Result does not fit into the object
8. MDCS_ERR_APPLICATION	3006	Error happened in application function
9. MDCS_ERR_OBJECT_EXIST	3007	Cannot continue, object exists
10. MDCS_ERR_DELETE_APPLICATION_OBJECT	3008	Cannot delete application object
11. MDCS_ERR_CANNOT_CONVERT_DATA	3009	Cannot perform data conversion
12. MDCS_ERR_CANNOT_SAVE_DATA	3010	Cannot save data
13. MDCS_ERR_EMPTY_QUERY	3011	Cannot execute empty query
14. MDCS_ERR_INCORRECT_QUERY	3012	Query is incorrect

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Virtual Callback Classes

18

In order to provide a consistent way to work with data, MDCStoreUtils provides the following virtual callback classes to users so that they can use them as an interface to define their own class definitions when using callbacks:

MDCStoreUtils also provides the MDCS_DBHandleSmartPtr class to deal with database handles.

Class MDCS_GroupInfoCallback

Overview

To retrieve multiple MDCS_ST_GroupInfo structure results which describes group information from database.

Public Methods

MDCS_GroupInfoCallback	Default constructor
~MDCS_GroupInfoCallback	Virtual destructor
ItemCount	Pure virtual function. To get the total number of groups.
GetNextInfo	Pure virtual function. To get group information from database record.

Private Methods

MDCS_GroupInfoCallback	Copy constructor
operator=	Assignment operator

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Function signature of MDCS_GroupInfoCallback

```
MDCS_GroupInfoCallback () {};
```

Default constructor to construct MDCS_GroupInfoCallback object.

```
virtual ~MDCS_GroupInfoCallback() {};
```

Virtual destructor destroys the MDCS_GroupInfoCallback object.

```
virtual ItemCount(LONGLONG lGroupCount) = 0;
```

To get a total number of groups.

Parameters

lGroupCount - number of groups counted

Return

Total number of groups.

```
virtual GetNextInfo(const MDCS_ST_GroupInfo& stGroupInfo)
= 0;
```

To get group information.

Parameters

stGroupInfo - A structure describes GroupInfo

Output

stGrouptInfo - A structure of MDCS_ST_GroupInfo contains information of the group

```
MDCS_GroupInfoCallback(MDCS_GroupInfoCallback &);
```

The default copy constructors and it is private.

```
const MDCS_GroupInfoCallback
&operator=(MDCS_GroupInfoCallback const &);
```

Assignment (**=**) operator reinitializes an existing MDCS_GroupInfoCallback object with new data.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Class MDCS_SaveBlobCallback

Overview

MDCS_SaveBlobCallback - callback for saving data blobs.

Public Methods

MDCS_SaveBlobCallback	Default constructor
~MDCS_SaveBlobCallback	Virtual destructor
NextChunk	Pure virtual function. To get the next chunk of data.
Done	Pure virtual function. Called when completely done.
GetPacketSize	Pure virtual function. To get size of transfer package.
GetTotalSize	Pure virtual function. To get total size of data transfer.
Error	Pure virtual function. Sent when there is an error.

Private Methods

MDCS_SaveBlobCallback	Copy constructor
operator=	Assignment operator

Function signatures of MDCS_SaveBlobCallback

```
MDCS_SaveBlobCallback () {};
```

Default constructor to initialize the MDCS_SaveBlobCallback.

```
virtual ~MDCS_SaveBlobCallback() {};
```

Virtual destructor

```
virtual BOOL NextChunk(
BYTE* pChunk,
const UINT& uSizeChunk,
UINT& uSizeRead) = 0;
```

Purpose

To get the next chunk of data.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

pChunk - pointer to chunk of data.
uSizeChunk - size that was requested
uSizeRead - actual size that was read

Output

pChunk - pointer to next chunk of data

```
virtual void Done() = 0;
```

Call this function when the saving process is completely done.

```
virtual UINT GetPacketSize() const = 0;
```

To get the size of the transfer package.

```
virtual LONGLONG GetTotalSize() const = 0;
```

To get the total size of data transferred.

```
virtual void Error(LPCSTR pErrorText) = 0;
```

Sent error text when there is an error.

Parameter

pErrorText - pointer to error text

Output

pErrorText - pointer to the sent error text

```
MDCS_SaveBlobCallback(MDCS_SaveBlobCallback &);
```

A default copy constructor.

```
const MDCS_SaveBlobCallback
&operator=(MDCS_SaveBlobCallback const &);
```

Assignment operator reinitializes the MDCS_SaveBlobCallback object to new data.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Class MDCS_GetBlobCallback

Overview

MDCS_GetBlobCallback - callback class for getting data blobs.

Public Methods

MDCS_GetBlobCallback	Default constructor
~MDCS_GetBlobCallback	Virtual destructor
NextChunk	Pure virtual function. To get the next chunk of data received from the database.
NextResult	Pure virtual function. Return size of the result.
GetPacketSize	Pure virtual function. To get the size of transfer package.
GetTotalSize	Pure virtual function. To get total size of data.
Done	Pure virtual function. Called when 100 percent complete.
Error	Pure virtual function. Sent when there is an error.

Private Methods

MDCS_GetBlobCallback	Copy constructor
operator=	Assignment operator

Function signature of MDCS_GetBlobCallback

```
MDCS_GetBlobCallback () {};
```

Default constructor

```
virtual ~MDCS_GetBlobCallback () {};
```

Virtual destructor

```
virtual BOOL NextResult(LONGLONG lSizeOfResult) = 0;
```

To get the size of the result.

Parameter

lSizeOfResult - Signed 64-bit integer to hold the result size.

Output

lSizeOfResult - the result size

```
virtual BOOL NextChunk(const BYTE* pChunk, UINT
uChunkSize) = 0;
```

To get the next chunk of data that will be received from database or file server.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

pChunk - pointer to the data
uChunkSize - size

Output

pChunk - pointer to the retrieved data.

```
virtual LONGLONG GetTotalSize() const = 0;
```

To get total size of data.

```
virtual UINT GetPacketSize() const = 0;
```

To get size of the transfer packet.

```
virtual void Done() = 0;
```

Call this function when getting data completely done.

```
virtual void Error(LPCSTR pErrorText) = 0;
```

The error text is sent when there is an error.

Parameter

pErrorText - pointer to error text

Output

pErrorText - pointer to the sent error text

```
MDCS_GetBlobCallback (MDCS_GetBlobCallback const &);
```

The default copy constructors.

```
const MDCS_GetBlobCallback
&operator=(MDCS_GetBlobCallback const &);
```

Assignment = operator reinitializes the MDCS_GetBlobCallback object to a new data.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Class MDCS_ProgressCallback

Overview

An abstract class of encapsulated functions to work with progress callback.

Public Methods

MDCS_ProgressCallback	Default constructor
~MDCS_ProgressCallback	Virtual destructor
Progress	Pure virtual function. To get percentage done and tell how many percent done.
Done	Pure virtual function. Called when 100 percent complete.
Error	Pure virtual function. Sent when there is an error.

Private Methods

MDCS_ProgressCallback	Copy constructor
operator=	Assignment operator

Function signature of MDCS_ProgressCallback

```
MDCS_ProgressCallback() {};
```

The default constructor to construct the MDCS_ProgressCallback object

```
virtual ~MDCS_ProgressCallback() {};
```

Virtual destructor to destroy dynamic data members.

```
virtual BOOL Progress(
int nPercentDone,
LPCSTR pProgressText,
BOOL bComplete)= 0;
```

To determine percentage progress.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

nPercentDone - between 0 and 100
pProgressText - string telling what is being done
bComplete - TRUE if 100% is really 100%

Output

bProgressText - string telling what is being done

```
virtual void Done()=0;
```

Called when completely done.

```
virtual BOOL Error(
LPCSTR pErrorText,
BOOL bWithContinue = FALSE)=0;
```

The error text is sent when there is an error. If *bWithContinue* is TRUE, error is not critical and process can be continued.

Parameters

pErrorText - pointer to error text
bWithContinue - to indicate whether the process can be continued

Output

pErrorText - pointer to error text

Return

FALSE - exist
TRUE - continue

```
MDCS_ProgressCallback(MDCS_ProgressCallback const &);
```

The default copy constructors.

```
const MDCS_ProgressCallback
&operator=(MDCS_ProgressCallback const &);
```

Assignment = operator reinitializes object MDCS_ProgressCallback to new data.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Class MDCS_GetDBResultsCCallback

Overview

An abstract class of encapsulated functions to work with database result callback.

Public Methods

MDCS_GetDBResultsCCallback	Default constructor
~MDCS_GetDBResultsCCallback	Virtual destructor
GetResultInfo	Pure virtual function. Called to get information about result structure (field names and types), all fields will be empty.
GetNextResult	Pure virtual function. Called to get next row of data.
Done	Pure virtual function. Called when 100 percent complete.
Error	Pure virtual function. Called when there is an error.
ProcessResultsInOriginalFormat	Pure virtual function. Called by function to check if results should be return in their original format.
SetResultNumber	Pure virtual function. To pass back to client number of results to be retrieved. Called to get number of results.

Private Methods

MDCS_GetDBResultsCCallback	Copy constructor
Operator=	Assignment operator

Function signature of MDCS_GetDBResultsCCallback

```
MDCS_GetDBResultsCCallback() {};
```

Default constructor to construct the MDCS_GetDBResultsCCallback.

```
virtual ~MDCS_GetDBResultsCCallback() {};
```

Virtual destructor

```
virtual BOOL GetNextResult(MDCS_QueryResults* pQueryRes)
= 0;
```

To get next result of data.

Parameter

pQueryRes - pointer to object MDCS_QueryResults

Output

pQueryRes - pointer to object MDCS_QueryResults

```
virtual void SetResultNumber(LONGLONG nCount) = 0;
```

To get number of results.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameter

nCount - Signed 64-bit integer to hold the number of result.

Output

nCount - the number of result

```
virtual void Error(LPCSTR pErrorText) = 0;
```

To send the error text when an error occurred.

Parameter

pErrorText - pointer to error text

Output

pErrorText - pointer to error text

```
virtual void Done() = 0;
```

Called when completely done.

```
virtual BOOL ProcessResultsInOriginalFormat() const = 0;
```

Called by function to check if results should be returned in their original format.

```
virtual BOOL GetResultInfo(MDCS_QueryResults* pQueryRes);
```

Called to get information about results structure (field names and types).

All fields will be empty.

Parameter

pQueryRes - pointer to MDCS_QueryResults

Output

pQueryRes - pointer to MDCS_QueryResults

Return

TRUE always

```
MDCS_GetDBResultsCCallback(MDCS_GetDBResultsCCallback
const &);
```

The default copy constructor.

```
const MDCS_GetDBResultsCCallback
&operator=(MDCS_GetDBResultsCCallback const &);
```

Assignment operator reinitializes the object MDCS_GetDBResultsCCallback to new data.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Class MDCS_QueryResults

Overview

An abstract class of encapsulated functions to work with QueryResults.

Public Methods

MDCS_QueryResults	Default constructor
~MDCS_QueryResults	Virtual destructor
GetStringValue	Pure virtual and Overloaded function. Function to get string result value of the field.
GetFloatValue	Pure virtual and Overloaded function. Function to get float result value of the field.
GetLongValue	Pure virtual and Overloaded function. Function to get integer result value of the field.
GetFieldType	Pure virtual and Overloaded function. Return data type of column.
GetColumnName	Pure virtual function. Returns name of the column.
GetColumnsCount	Pure virtual function. Returns number of columns.

Private Methods

MDCS_QueryResults	Copy constructor
operator=	Assignment operator

Function signature of MDCS_QueryResults

```
MDCS_QueryResults() {};
```

Default constructor to initialize MDCS_QueryResults object.

```
virtual ~MDCS_QueryResults(){};
```

Overridable destructor.

```
virtual LPCSTR GetStringValue(int nCol) const = 0;
```

Overloaded function to get string result value of the field.

Parameter

nCol - column number

Output

A pointer to a string result.

```
virtual LPCSTR GetStringValue(LPCSTR strColumnName) const
= 0;
```

Overloaded function to get string result value of the field.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameter

strColumnName - a pointer to column name of the field of result

Return

A pointer to string value result.

```
virtual float GetFloatValue(int nCol) const = 0;
```

Overloaded function to get float result value of the field.

Parameter

nCol - column number or field number of result

Return

A float value result.

```
virtual float GetFloatValue(LPCSTR strColumnName) const =
0;
```

Overloaded function to get float result value of the field.

Parameter

strColumnName - pointer to column name or field name.

Return

A float result value of the field.

```
virtual LONGLONG GetLongValue(int nCol) const = 0;
```

Overloaded function to get integer value of the field.

Parameter

nCol - column number, field number of result

Return

A signed integer value result of the field.

```
virtual LONGLONG GetLongValue(LPCSTR strColumnName) const
= 0;
```

Overloaded function to get integer result value of the field.

Parameter

strColumnName - pointer to a column name or field name

Return

A signed integer value that is result of the field.

```
virtual MDCS_E_ColumnType GetFieldType(int nCol) const =
0;
```

Overloaded function to get type of column.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameter

nCol - column number or field number

Return

Type of the column.

```
virtual MDCS_E_ColumnType GetFieldType(LPCSTR
strColumnName) const = 0;
```

Overloaded function to get type of column.

Parameter

strColumnName - pointer to column name

Return

The enum field type of column.

```
virtual LPCSTR GetColumnName(int nCol) const = 0;
```

To get name of the column.

Parameter

nCol - column number or field number

Return

Name of the column.

```
virtual int GetColumnsCount() const = 0;
```

To get number of columns.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Class MDCS_ImportMeasurementSet

Overview

A data source to import measurement set data into the database.

Public Methods

MDCS_ImportMeasurementSet	Default constructor
~MDCS_ImportMeasurementSet	Virtual destructor
GetMeasurementAttributeNumber	Pure virtual function. To get the number of measurement attributes that applied to a cell, not including ones that are in MDCS_ST_CellDescription.
GetCellCount	Pure virtual function. To get the total number of rows to be inserted.
GetMeasurementAttributeInfo	Pure virtual function. To get measurement column information.
GetCellDescription	Virtual function. To get the cell description for each row that we insert.
GetMeasurementValue	Overloaded and pure virtual function. To get measurement set value to insert.
GetNumberRowsInStep	Pure virtual function. To get max number of rows to be inserted in one step.
StatusWhenDone	Virtual function. To set assay status when import done.
StatusWhileProccessing	Virtual function. To set a status of assay while importing.

Private Methods

MDCS_ImportMeasurementSet	Copy constructor
operator=	Assignment operator

Function signatures of MDCS_ImportMeasurementSet

```
MDCS_ImportMeasurementSet() {};
```

Default constructor to initialize MDCS_ImportMeasurementSet object.

```
virtual ~MDCS_ImportMeasurementSet() {};
```

Virtual destructor.

```
virtual UINT GetMeasurementAttributeNumber() const = 0;
```

To get number of measurement attributes that applied to a cell, not including ones that are in MDCS_ST_CellDescription.

```
virtual BOOL GetMeasurementAttributeInfo(
UINT uColumnNum,
LONGLONG& nAttributeID) = 0;
```

To get measurement column information.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

uColumnNum - Column number
nAttributeID - an ID of a global measurement attribute

Output

Measurement column information.

```
virtual UINT GetCellCount() const = 0;
```

To get total number of rows to be inserted.

```
virtual BOOL GetCellDescription(
UINT uCellIndex,
MDCS_ST_CellDescription& stCellDescription) = 0;
```

To get cell description for each row that we insert.

Parameters

nCellIndex - index of the cell
stCelDescription - structure of Cell description

Output

A structure of cell description.

```
virtual BOOL GetMeasurementValue(
UINT uCellIndex,
UINT uAttrIndex,
LPSTR pszData,
UINT uSize) const = 0;
```

To get measurement set value to insert.

Parameters

uCellIndex - index of cell
uAttrIndex - attribute index
pszData - pointer to measurement data
uSize - size of the data, use MDCS_MAX_STRING_VALUE for it.

```
virtual BOOL GetMeasurementValue(
UINT uCellIndex,
UINT uAttrIndex,
float** pfData) const = 0;
```

To get float measurement set value to insert.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameter

pfData - pointer to float value that will be inserted

```
virtual BOOL GetMeasurementValue(
UINT uCellIndex,
UINT uAttrIndex,
int** pnData) const = 0;
```

To get integer measurement set value to insert

Parameter

pnData - pointer to int value that will be inserted

```
virtual UINT GetNumberRowsInStep() const = 0;
```

To get max number of rows to be inserted in one step.

```
MDCS_ImportMeasurementSet (MDCS_ImportMeasurementSet
const &);
```

Copy constructor.

```
const MDCS_ImportMeasurementSet
&operator=(MDCS_ImportMeasurementSet const &);
```

Assignment operator reinitializes MDCS_ImportMeasurementSet object to new data.

```
virtual MDCS_E_AssayMarkStatus StatusWhenDone(
) const { return eMarkNormal;};
```

```
To set assay status when import is done.
```

```
virtual MDCS_E_AssayMarkStatus StatusWhileProccessing(
)  const { return eMarkRemove;};
```

```
To set a status of assay while importing.
```

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Class MDCS_ImportPlateLayout

Overview

A data source to import plate layout data to the database.

Public Methods

MDCS_ImportPlateLayout	Default constructor
~MDCS_ImportPlateLayout	Virtual destructor
GetNumberWells	Pure virtual function. To get the number of columns in the data source.
GetNumberGroups	Pure virtual function. To get the number of groups on a plate.
GetNumberCompounds	Pure virtual function. To get the number of compounds on a plate.
GetWellInfo	Pure virtual function. To get the well description.
GetCompound	Pure virtual function. To get the compound description.
GetConcentration	Pure virtual function. To get the concentration of compound.
GetUnit	Pure virtual function. To get the unit data.
GetWellGroup	Pure virtual function. To get the well group.

Private Methods

MDCS_ImportPlateLayout	Copy constructor.
operator=	Assignment operator

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Function signature of MDCS_ImportPlateLayout

```
MDCS_ImportPlateLayout() {};
```

Default constructor

```
virtual ~MDCS_ImportPlateLayout() {};
```

Virtual destructor

```
virtual UINT GetNumberWells() const = 0;
```

To get the number of columns in the datasource.

```
virtual UINT GetNumberGroups() const = 0;
```

To get the number of groups on a plate.

```
virtual UINT GetNumberCompounds() const = 0;
```

To get number of compounds on a plate.

```
virtual BOOL GetWellInfo(
UINT uWellIndex,
MDCS_ST_WellInfo& stWellInfo) const = 0;
```

To get the Well description.

Parameters

uWellIndex - well index
stWellInfo - Well information structure

Output

stWellInfo - well information structure

```
virtual BOOL GetCompound(
UINT uWellIndex,
UINT uCompoundIndex,
LPSTR pszData,
UINT uSize) const = 0;
```

To get the compound description.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

uWellIndex - well index of compound
uCompoundIndex - compound index
pszData - pointer to string of compound description
uSize - size of the compound description string

Output

pszData - pointer to compound description

```
virtual BOOL GetConcentration(
UINT uWellIndex,
UINT uCompoundIndex,
float** fConc) const = 0;
```

To get concentration of a compound in a well.

Parameters

uWellIndex - well index
uCompoundIndex - compound index
fConc - pointer to float value to be inserted

Output

Concentration of the compound.

```
virtual BOOL GetUnit(
UINT uWellIndex,
UINT uCompoundIndex,
LPSTR pszData,
UINT uSize) const = 0;
```

To get unit of data.

Parameters

uWellIndex - well index
uCompoundIndex - compound index
pszData - pointer to data
uSize - size of data string

Output

Unit of data.

```
virtual BOOL GetWellGroup(
UINT uWellIndex,
UINT uGroupIndex,
LPSTR pszGroupName,
UINT uSize) const = 0;
```

To get the Well group description.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Parameters

uWellIndex - well index

uGroupIndex - group index

pszGroupName - group name

usize - size of group name string

Output

pszGroupName - group name description

```
MDCS_ImportPlateLayout(MDCS_ImportPlateLayout const &);
```

Default copy constructor.

```
const MDCS_ImportPlateLayout
&operator=(MDCS_ImportPlateLayout const &);
```

Assignment operator reinitializes MDCS_ImportPlateLayout object to new data.

Class MDCS_DBHandleSmartPtr

Overview

MDCS_DBHandleSmartPtr is not a pure virtual callback class. It is a smart pointer that is a wrapper around HDBHANDLE to deal with HDBHandles. It initializes and destroys dbhandle, so you do not need to destroy a database handle after use. However, it can be used only on the stack.

Data Member

m_hdbHandle	A HDBHANDLE database handle.

Public Methods

MDCS_DBHandleSmartPtr	Constructor to initialize user login information.
~MDCS_DBHandleSmartPtr	Virtual Destructor
GetHandle	Return the HDBHANDLE database handle.

Private Methods

MDCS_DBHandleSmartPtr	Default constructor
operator=	Overridden = assignment operator.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Function signature of MDCS_DBHandleSmartPtr

```
MDCS_DBHandleSmartPtr() {};
```

Default constructor to construct the MDCS_DBHandleSmartPtr object.

```
MDCS_DBHandleSmartPtr& operator=(

const MDCS_DBHandleSmartPtr &stConnection

);
```

Assignment operator reinitializes the MDCS_DBHanldeSmartPtr to a new data.

```
MDCS_DBHandleSmartPtr(const MDCS_ST_UserLogin&
stUserLogin,

BOOL bSilentMode = FALSE, BOOL bConnectAsAppUser = TRUE);

MDCS_DBHandleSmartPtr(LPCTSTR pszConnectionString,

BOOL bSilentMode = FALSE);
```

Constructor to initialize the user login information.

Parameters

bSilentMode - Indicates if connection should be in silent mode.
bConnectAsAppUser - Indicates if the user is an application user.

```
virtual ~MDCS_DBHandleSmartPtr();
```

Virtual destructor.

```
HDBHANDLE GetHandle();
```

Return the database handle of this object.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Class MDCS_CL_ImportDS

Public Methods

MDCS_CL_ImportDS	Constructor to initialize measurement column database name.
GetAttributeDBName	Virtual function. To get measurement column database name.
GetTableName	Virtual function. To get measurement table name.

Function signature of MDCS_CL_ImportDS

To get Attribute database name

```
virtual LPCTSTR GetAttributeDBName(UINT uColumnNum) const
= 0;
```

Parameters

uColumnNum - column number

Output

Attribute Name

To get attribute table name

```
virtual LPCTSTR GetTableName() const = 0;
```

Parameters

None

Returns

Table name

```
};
```

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Class MDCS_GetProgressStatus

Public Methods

MDCS_GetProgressStatus	Default constructor
~MDCS_GetProgressStatus	Virtual destructor
GetProgress	Virtual function to get current progress.
GetProgressText	Virtual function to get current progress text.
GetTitleText	Virtual function to get current progress title text.
GetStatus	Virtual function to get a status.
GetError	Virtual function to get an error description.

Private Methods

MDCS_GetProgressStatu s	To hide the default copy constructors.

Function signature of MDCS_GetProgressStatus

```
MDCS_GetProgressStatus() {};
```

Default constructor to initialize MDCS_GetProgressStatus object.

```
virtual ~MDCS_GetProgressStatus() {};
```

To get current progress.

```
virtual UINT_PTR GetProgress() const = 0;
```

To get current progress.

```
virtual LPCSTR GetProgressText() const = 0;
```

To get current progress.

```
virtual LPCSTR GetTitleText() const = 0;
```

To get current progress.

```
virtual E_ProgressStatus GetStatus() const = 0;
```

To get a status.

```
virtual LPCSTR GetError() const = 0;
```

To get an error description.

```
MDCS_GetProgressStatus(MDCS_GetProgressStatus const &);
      const MDCS_GetProgressStatus


      &operator=(MDCS_GetProgressStatus const &);
```

To hide the default copy constructors

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Class MDCS_CL_BlobLocationCB

This is a class to process an object count.

Public Methods

MDCS_CL_BlobLocationCB	Constructor to initialize object count.
~MDCS_CL_BlobLocationCBr	Virtual Destructor

Private Methods

MDCS_CL_BlobLocationCB	To hide the default copy constructor.
operator=	Overridden = assignment operator.

Return

```
    virtual void NextResult(const MDCS_ST_BlobLocation&) = 0;
```

class MDCS_CL_BlobInfoCB

This is a class to process blob information.

Public Methods

MDCS_CL_BlobInfoCB	Constructor to initialize object count.
~MDCS_CL_BlobLocationCBr	Virtual Destructor

Private Methods

MDCS_CL_BlobInfoCB	To hid the default copy constructor
operator=	Overridden = assignment operator.

Return

virtual void NextResult(const MDCS_ST_BlobInfo&) = 0;

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

19 Usage Examples

This chapter provides some examples of how to use functions in MDCStoreUtils API to do the following:

- Make a connection to a datasource and get a database handle.
- Get plate, site, and image information.
- Show database error messages.
- Derive from pure virtual callback classes.

How to Connect to the Database and Get a Database Handle

In order to work with the MDCStore™ database, you need to have a database handle, which is a parameter required in most of the functions in MDCStoreUtils API. You can get a database handle easily by calling the function **MDCS_CONNECTION_GetDBHandle**. In order to call the function, you need to have user login information to pass to this function to create a database connection and return a database handle. When filling the user login information, the following members of MDCS_ST_UserLogin need to be specified:

- szUserName
- szPassword
- szDSN
- szDatabase - optional. It is used when an ODBC datasource was set up without indicating database name.

You can fill out the information above manually or by calling function **MDCS_CONNECTION_GetDetails**.

The function **MDCS_CONNECTION_GetDetails** displays a dialog where you can enter the login information.

Note: Comments within the code examples are shown in a green font.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

The following example shows how to get user information and a database handle using these functions:

```
MDCS_ST_UserLogin stUserLogin ;
//holds user login information
   HDBHANDLE hDBHandle;
//database handler
//Call the following function to gather login information
from user and
//store information in stUserLogin structure.

if(!MDCS_CONNECTION_GetDetails(&stUserLogin,NULL,NULL,"MD
CS Example",NULL,FALSE))
  {
//show database error, will discuss in next section
      ShowDBDllError(“Function Fail.\nReason:");
//implemented by user, its definition is below
  }
//Another way is to initialize the user name, password and
datasource name in stUserLogin
//manually and use it to connect to database.
//For example,
//strcpy(stUserLogin.szUserName, "sa");
//strcpy(stUserLogin.szPassword, "sa");
//strcpy(stUserLogin.szDSN, "IX_MDCStoreDemo");
   cout << "Next! We are getting database handle" << endl;
 //Call function in MDCStoreUtilApi to get database
handle
   hDBHandle = MDCS_CONNECTION_GetDBHandle(stUserLogin);
```

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Another way to get a database handle is to use the **MDCS_DBHandleSmartPtr** object. Using this object you don’t have to destroy the database handle when you are finished using it because the destructor of **MDCS_DBHandleSmartPtr** takes care of it. This is demonstrated in the **ExecMain** function below:

```
void ExecMain(const MDCS_ST_UserLogin& stUserLogin,
LONGLONG lPlateID)

{

//get database handle

   MDCS_DBHandleSmartPtr ptrDB(stUserLogin);

//check if we got a valid connection

   if (ptrDB.GetHandle())

   {

//get images for a plate that have ID 18

      BOOL  bReturn = GetImages(ptrDB.GetHandle(),
lPlateID);

      if (!bReturn )

         ShowDBDllError("Function failed.\nReason:");

   }

   else

   {

      cout << "Execution cancelled." << endl;

   }


   int ch;

   _cputs( "\n\nPress any key to exit." );

   ch = _getch();

   _cputs( "\r\n" );

}
```

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

How to Use Error Handling

There is a function call to **ShowDBDllError** to display an error message in the first example above. **ShowDBDllError** is a typical way to use the error-handling functions in MDCStoreUtils API to display the database error whenever a MDCStoreUtils API function fails. **ShowDBDllError** calls **MDCS_GetLastErrorMsg** to get the last database error and displays the error in a console window. Here is an example of how to call **MDCS_GetLastErrorMsg** to do error handling:

```
BOOL  ShowDBDllError(const CString& strErrorExtra )

{

   char szError[MDCS_MAX_ERR_SIZE] = "";

  if(MDCS_GetLastErrorMsg(szError, MDCS_MAX_ERR_SIZE) ==
MDCS_ERR_SUCCESS)

       return FALSE;

   else

  {

    CString strErr;

    strErr.Format("%s\n%s", strErrorExtra, szError);

    cout << strErr << endl;

    return TRUE;

  }

   return FALSE;

}
```

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

How to Get Plate Information

Once you have a database handle, you can access any data in the MDCStore database, such as plate information, site information belonging to a specific plate, or an image from the database. Below is a function that calls **MDCS_PLATE_GetInfo** to get plate information:

```
//=====================================================
//FUNCTION: GetPlateInfo
//PURPOSE:  Get plate information and print out the plate
information
//Parameters
//    hDBHandle - database handle
//    lPlateID - a plate ID
//RETURN Return false - if fails
//=====================================================
BOOL  GetPlateInfo(HDBHANDLE hDBHandle, LONGLONG
lPlateID)
{
   MDCS_ST_PlateInfo st_PlateInfo;
//to hold the plate information
   if(!MDCS_PLATE_GetInfo(hDBHandle, lPlateID,
st_PlateInfo))
   {
//You can show DATABASE error here by calling
//ShowDBDllError("Function fails."); or let the function,
where GetPlateInfo
//function is called handle the error, or the main
program.
      return FALSE;
   }
   else
   {
// Print out plate information
      cout << "The plate information " << endl;
      cout << "Plate name:\t" << st_PlateInfo.szPlateName
<< endl;
      cout << "Xwells:\t"  << st_PlateInfo.nXWells <<
endl;
     cout << "Ywells:\t" << st_PlateInfo.nYWells << endl;
```

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

```
      cout << "Acquisition:\t" << st_PlateInfo.szAcqName
<< endl;
      cout << "Barcode:\t" << st_PlateInfo.szBarcode <<
endl;
      cout << "Creator:\t" << st_PlateInfo.szCreator <<
endl;
      cout << "Description:\t" << st_PlateInfo.szDesc <<
endl;
      cout << "\n\n" ;
   }
   return TRUE;
}
```

In the previous example the function returns FALSE when any function call fails, and we did not display the error here. You can handle the error here by calling **ShowDBDllError**, but we intend to handle the error in the function that calls **GetPlateInfo**, so we simply return false if there is an error. It is up to you to decide where to handle the error. If the error is handled in this function then you do not need to handle the error in the function that called it.

How to Get All Sites Per Plate

MDCStoreUtils API provides the function **MDCS_PLATE_GetSitesByPlate** to get all sites per plate. It uses a callback function **GetNextResult** in the pure virtual class **MDCS_GetDBResultsCCallback** to process the result. Therefore, first you need to derive the **MDCS_GetDBResultsCCallback** class and provide the definition to the **GetNextResult** method to process results. Let’s create the **CGetSitesByPlate_CallBack** callback class inherited from **MDCS_GetDBResultsCCallback** to process and get all site IDs of a plate returned from the database by the **MDCS_PLATE_GetSitesByPlate** function. Deriving virtual callback classes is discussed in next section.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Deriving the MDCS_GetDBResultCCallback Class

When deriving any pure virtual callback class in MDCStoreUtils, the way that you implement the class depends on what you intend to get from the database. The data returned from the database determines the data member variable to hold the result in the derived class.

The function **MDCS_PLATE_GetSitesByPlate** in MDCStoreUtils API gets all sites per plate. It returns records with columns:

```
ID - as SITE_ID

BATCH_ID - as batch ID

      X_WELLS - number of X wells

Y_WELLS - number of Y wells

      X_POSITION - X position within a well

Y_POSITION - Y position within a well

TO_DELETE - flag of deletion
```

What is needed is only site IDs belong to the plate, so in this case the data member is an array of siteIDs **m_plarrSiteIDs** (see below in GetSitesByPlate_CallBack.h). The function **GetNextResult** (see below in GetSitesByPlate_CallBack.cpp) gets the next row of data. When implementing this function, you just need to call the right function of pointer **pQueryRes** that points to the **MDCS_QueryResults** object to get the data you want and to put it in the data member variable.

CGetSitesByPlate_CallBack.h and **GetSitesByPlate_CallBack.cpp** below show how to implement the derived class in this case.

```
//*******************************************************
**************

// Copyright (c) 2005 Molecular Devices

// All rights reserved.

//*******************************************************
*************

// Module:  GetSitesByPlate_CallBack.h

// Purpose: Provide functions to process data and get all
Site IDs belonging to a plate

//

/////////////////////////////////////////////////////////
/////////////

#pragma once

#include "mdcstoreutilsapi.h"

#include <afxtempl.h>


class CGetSitesByPlate_CallBack :

   public MDCS_GetDBResultsCCallback
```

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

```
{
public:
   CGetSitesByPlate_CallBack(CArray<LONGLONG, LONGLONG>*
arrSiteID);


   virtual ~CGetSitesByPlate_CallBack(void){}
   //called to get next row of data
   virtual BOOL  GetNextResult(MDCS_QueryResults*
pQueryRes) ;
   //called to get number of results
   virtual void SetResultNumber(LONGLONG nCount)
   {};
   //called when there is an error
   virtual void Error(LPCSTR pErrorText)
   {
      return;
   }
   // called when completely done
   virtual void Done()
   {
      return;
   }
   //called to check if results should be returned in
their original format
   virtual BOOL  ProcessResultsInOriginalFormat() const
   {
```

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Using the Callback CGetSitesByPlate_Callback

Once you have the callback class **CGetSitesByPlate_CallBack**, you are ready to call the **MDCS_PLATE_GetSitesByPlate** function to get all siteIDs belonging to a plate. The following function is an example to illustrate how to get all siteIDs belonging to a specific plate:

```
//Get all site IDs belonging to a plate

BOOL  GetAllSitesPerPlate(HDBHANDLE hDBHandle, lPlateID)

{

      CArray<LONGLONG, LONGLONG> larrSiteIDs ;

//hold all site IDs of the plate

      CGetSitesByPlate_CallBack pResult(&larrSiteIDs);

//Callback

//Call function in MDCStoreUtils

     if(!MDCS_PLATE_GetSitesByPlate(hDBHandle, lPlateID,
&pResult))

      {

         return FALSE;

      }

      return TRUE;

}
```

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Deriving the MDCS_GetBlobCallback Class

Similar to getting siteIDs per plate above, getting images requires that you provide a callback class. In this case the callback class is derived from **MDCS_GetBlobCallback.** Below is a typical way to implement the callback class you need. The **CBlobCallbackEx** class allows writing blob data directly to file using a file handle. The most important method in this class is **NextChunk**. It is used to get the next chunk of data from the source (database or file server) and write it to a file directly.

```
//*******************************************************
**************

// Copyright (c) 2005 Molecular Devices

// All rights reserved.

//*******************************************************
**************

//Module:  CBlobCallbackEx.cpp

//PURPOSE: Extension of the class CBlobCallback, it

//         allows you to save data right into a file

/////////////////////////////////////////////////////////
//////////////

#include "stdafx.h"

#include ".\blobcallbackex.h"

// Constructor to initialize the object

CBlobCallbackEx::CBlobCallbackEx( HANDLE file, LONGLONG
uTotalSize, LPCSTR pszFileName, UINT uChunkSize /*=
DEFAULT_CHUNK_SIZE */) :

    m_file(file), m_uChunkSize(uChunkSize),
m_uTotalSize(uTotalSize), m_strFileName(pszFileName)

{

   m_lCurSize = 0;

}

//=====================================================

// FUNCTION: NextResult

// PURPOSE:  Next result is retrieved, returns size of the
result

//=====================================================

BOOL  CBlobCallbackEx::NextResult(LONGLONG lResultSize)

{

    return TRUE;

}
```

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

```
//==========================================================
==============// FUNCTION: NextChunk
// PURPOSE:  Next chunk of data that will be received from
database
//
// PARAMETERS
//       pChunk - data
//       uChunkSize - size of data to be retrieved
// RETURN : FALSE - if error occurs
//==========================================================
==============BOOL  CBlobCallbackEx::NextChunk(const
BYTE* pChunk, UINT uChunkSize )
```

Using the CBlobCallbackEx Callback Class

Having a callback class CBlobCallbackEx ready, you can call the function **MDCS_BLOB_Get** in MDCStoreUtils to get image data.

```
MDCS_BLOB_Get(HDBHANDLE hHandle,
LONGLONG lBlobID,
MDCS_E_BLobType pBlobType,
MDCS_GetBlobCallback * pCallBack)
```

The **GetImages** function below illustrates the use of the **CBlobCallbackEx** callback class and the **MDCS_BLOB_Get** function to write image data to a file on the C drive.

```
//FUNCTION : GetImages
//PURPOSE:   Get all images from database using image ID
and save them to drive C
//Parameters
//    hDBHandle - database handle
//    larrImageIDs - array of ImageIDs of a specific site
//RETURN Return FALSE if fails
//=====================================================
BOOL  GetImages(HDBHANDLE hDBHandle, CArray<LONGLONG,
LONGLONG>& larrImageIDs)
{
//Create file handle to write to it
    MDCS_E_BlobType eBlobType = MDCS_eBlobSiteImage;
    LONGLONG lTotalSize = 0;
    CString strFileNameTemplate = "Image_%d.%s";
```

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

```
    CString strFileNameTemplate2 = "Image_%d.jpg";
    CString strFileName;
//get each image from database to C drive
    for ( int nCount = 0; nCount < larrImageIDs.GetSize();
nCount++ )
    {
//get image information and use information about
original file extension to
//create files on disk
      MDCS_ST_BlobInfo stBlobInfo;
      if (!MDCS_BLOB_GetInfo(hDBHandle, &stBlobInfo,
                             larrImageIDs.GetAt(nCount),
MDCS_eBlobSiteImage))
         return FALSE;
      if (!stBlobInfo.lBlobID)
      {
         cout << "Image with ID " <<
larrImageIDs.GetAt(nCount) << " is not
                 found in the database" << endl;
         continue;
      }
      CString strExt;
      if (!strlen(stBlobInfo.szImportExtension))
         strFileName.Format(strFileNameTemplate2,nCount);
      else
         strFileName.Format(strFileNameTemplate,nCount,
stBlobInfo.szImportExtension);
HANDLE hFile;
      //create image file
      hFile = CreateFile(strFileName,
                     FILE_WRITE_DATA,
                     FILE_SHARE_WRITE,
                     NULL,
                     CREATE_ALWAYS,
                     FILE_ATTRIBUTE_NORMAL,
                     NULL);
```

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

```
      if (hFile == INVALID_HANDLE_VALUE)
      {
         cout << "Could not open file." << endl;
      }
      CBlobCallbackEx pCallBack(hFile, lTotalSize,
strFileName , 1024*16);
      //1024*16 is default size
      if(!MDCS_BLOB_Get(hDBHandle, larrImageIDs[nCount],
eBlobType , &pCallBack))
      {
         CloseHandle(hFile);
         return FALSE;
      }
     cout <<  "Image " << nCount + 1 << " saved into " <<
strFileName.GetBuffer()<<
                endl;
      CloseHandle(hFile);
    }
    return TRUE;
}
```

The above examples demonstrate only some of the MDCStoreUtils API functions used to access the MDCStore database. To see how all the functions and classes discussed above are used in an application, you can examine the application MDCStoreImageExample attached with this documentation.

MDCStoreImageExample is a console application that demonstrates how to use functions in MDCStoreUtils API to get plate, site information, get images of from site, show database error messages, and use callback classes. It accepts the following parameters that are displayed if you call the application with an empty command line:

```
-u <User name>
-p <Password>
-dsn <ODBC datasource name>
-database <database name> - this parameter is optional
-plate <plate ID> - this parameter is optional
```

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

The application uses most of functions and classes discussed in this section. Always call **MDCStoreUtils_Init** to initialize the MDCStore interface before calling any other functions in MDCStoreUtils API and call **MDCStoreUtils_Finished** to detach the MDCStoreUtils interface when exiting the application.

The following is a list of the main source and header files used in the application:

- BlobCallbackEx.h
- GetImageRecord_CallBack.h
- GetSitesByPlate_CallBack.h
- MDCStoreImageExample.h
- BlobCallbackEx.cpp
- GetSitesByPlate_CallBack.cpp
- GetImageRecord_CallBack.cpp
- MDCStoreImageExample.cpp - main application

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

Index

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.

The MDCStore™ SDK and associated documentation are copyrighted and property of Molecular Devices, LLC. Use of the MDCStore™ API (Application Programming Interface) and acceptance of the associated documentation is prohibited without the express written consent of Molecular Devices, LLC, which consent shall limit the scope of use of the MDCStore™ technology, shall be personal to the licensed user, and shall be non-transferable.