

Technical Documentation Application Software

***Universal Database Library
(.hsl)***

Table of contents

1	Introduction	3
1.1	Purpose of this document	3
1.2	Definitions and Abbreviations	3
1.3	References.....	3
2	Concept.....	4
2.1	System structure	4
2.2	Previous .hsl-Library	5
2.3	Namespace	6
2.4	Variables and constants.....	6
2.5	Used software and versions.....	6
2.6	COM-Objects	7
2.7	Used / included standard libraries.....	7
3	hsl-library.....	8
3.1	Supported DBMS	8
3.2	Connection to DBMS (Drivers / Providers)	8
3.3	Functions of hsl-library	9
3.3.1	Library_Init	9
3.3.2	Library_DeInit.....	10
3.3.3	DBConnectionOpen	10
3.3.4	DBConnectionClose.....	10
3.3.5	TransactionBEGIN	10
3.3.6	TransactionCOMMIT.....	11
3.3.7	TransactionROLLBACK	11
3.3.8	ExecuteSELECT	11
3.3.9	GetFieldFromArrayOfSELECT.....	12
3.3.10	GetFieldFromArrayOfSELECT_Record.....	12
3.3.11	ExecuteUPDATE	12
3.3.12	ExecuteDELETE	13
3.3.13	ExecuteINSERT_OneRecord_ID.....	13
3.3.14	ExecuteSTATEMENT	13
3.4	Data-Types.....	14
4	Error handling	14
5	Concept of wrapper-library	15

1 Introduction

This technical software documentation is for internal use of HAMILTON Bonaduz AG software-engineers of group ASW.

1.1 Purpose of this document

The document is to describe the „Universal Database Library“ on a technical level.

1.2 Definitions and Abbreviations

.hsl	Hamilton Standard Language
.smt	Submethod library
ASE	Application Software Engineer
ASW	Application Software
DBMS	Database Management System
DB	Database

1.3 References

- Programmers Manual: 62404303_ProgrammersManual_VENUS_one.pdf
- AAW: 650336 (V 3.3)
- AAW: 650410 (V 1.0)
- AAW: 650338 (V 3.1)
- AAW: 650417 (V 1.0)
- AAW: 652310 (V 1.1)
- AAW: 652322 (V 1.2)

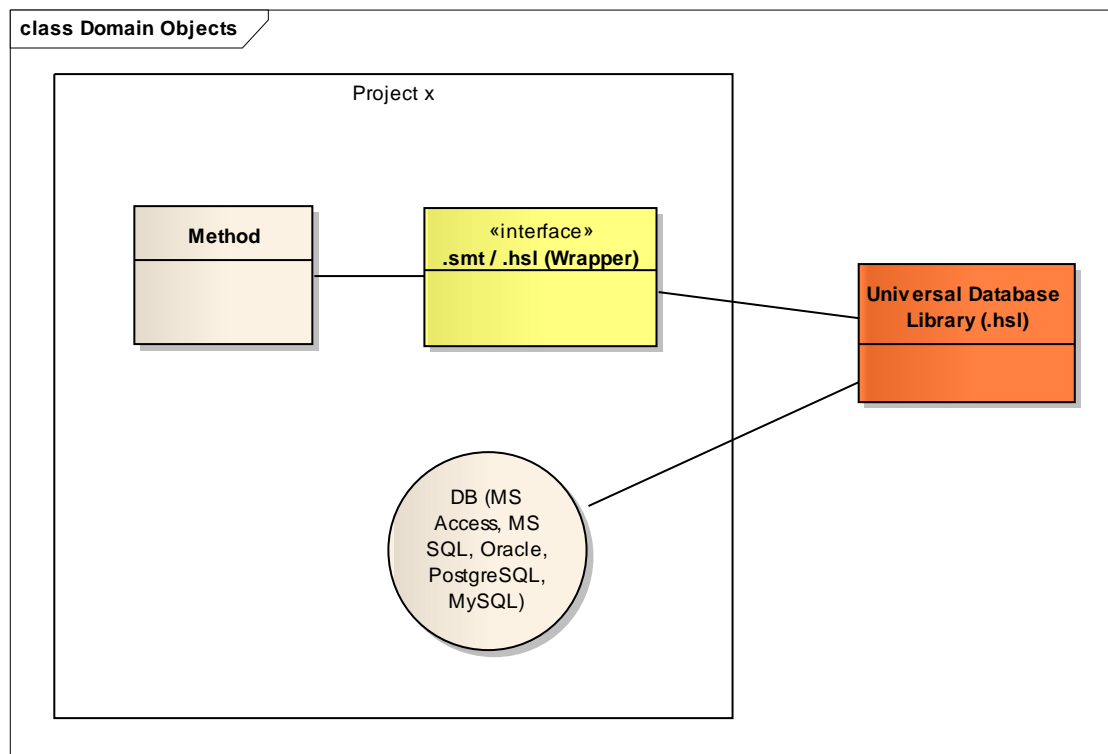
2 Concept

Basically, the concept of implementation is a method which uses a project specific wrapper-interface (.smt / .hsl) to access the low-level hsl-library for interacting with the DBMS.

- The project has implemented a wrapper-library for specific support of the methods (example: WrapperTest.med)
- The hsl-library is for general use and supports different DBMS

2.1 System structure

The system concept is shown in graphic below.



Project:	Universal Database Library	HAMILTON Bonaduz AG	2010-01-22	Site 5 of 15
Title:	Technical Documentation	Doc. Name:	SRS_UniversalDatabaseLibrary.doc	Version: 0.1

2.2 Previous .hsl-Library

The "old" hsl-library (until 01.2010) has implemented several public functions:

Function	Description	Input	Return / Output
Init	Initializes objects, semaphores and connects to the DBMS.	- DB Server Type - DB Name - DB Host Name - DB User Name - DB Password - Trace level for library traces	Return: hslTrue = no error by initializing hslFalse = error while initializing
Delnit	De-Initializes objects, semaphores and closes connection to DBMS	-	Return: hslTrue
SQLExecute_START	Waits for semaphore (lock)	-	Return: hslTrue
SQLExecute_STOP	Releases semaphore (unlock)	-	Return: hslTrue
SQLExecute_UPDATE	Executes a SQL update or delete statement	- SQL command: Update / Delete	Return: number of affected records
SQLExecute_INSERT	Executes a SQL insert statement	- SQL command: Insert - Sequence name (for Oracle / PostgreSQL)	Return: -1 insert failed / no error: primary key ID of inserted record
SQLExecute_SELECT	Executes a SQL select statement Opens record set (fetch)	- SQL command: Select	Return: hslTrue = one ore more matching records found hslFalse = no matching record found
SQLExecute_SELECT_Read	Reads first / next record of fetched records	-	Return: hslTrue = first, next record found / hslFalse = no record found

Project:	Universal Database Library	HAMILTON Bonaduz AG	2010-01-22	Site 6 of 15
Title:	Technical Documentation	Doc. Name:	SRS_UniversalDatabaseLibrary.doc	Version: 0.1

Function	Description	Input	Return / Output
SQLExecute_SELECT_Field	Reads a specific field of fetched record - Date is converted to string - Decimal is converted to integer or float - NULL (data type) is converted to empty string	- Name of field	Return: (converted) Value from field
SQLString_ActuallyDateTime	Creates a string with actual date and time depending on DBMS	-	Return: created string with date and time (unknown DBMS, return: "")
SQLString_CleanUp_SpezialCharacters_for_Strings	Cleans up characters for real syntax of a SQL statement. (character = "" --> character = "")	SQL command	Return: cleaned SQL statement

2.3 Namespace

Namespace: UNIVERSAL_SQL_DATABASE

2.4 Variables and constants

See declaration in hsl-library "Universal_SQL_Database_Library.hsl".

Special:

```
static const variable _DATABASE_CONNECTION_TIMEOUT(15);
```

- Number of seconds to wait for a **connection** to open, before canceling the attempt and generate an error (default of object is 15)

```
static const variable _DATABASE_COMMAND_TIMEOUT(60);
```

- Number of seconds to wait while attempting to **execute a command**, before canceling the attempt and generate an error.

2.5 Used software and versions

Standard Venus SW.

Project:	Universal Database Library	HAMILTON Bonaduz AG	2010-01-22	Site 7 of 15
Title:	Technical Documentation	Doc. Name:	SRS_UniversalDatabaseLibrary.doc	Version: 0.1

2.6 COM-Objects

Objects	Information
ADODB.Connection	http://www.w3schools.com/ado/ado_ref_connection.asp
ADODB.Recordset	http://www.w3schools.com/ado/ado_ref_recordset.asp
ADODB.Command	http://www.w3schools.com/ado/ado_recordset.asp
MSScriptControl.ScriptControl	http://support.microsoft.com/kb/184740/en-us?fr=1

2.7 Used / included standard libraries

HSLStrLib.hsl - Standard Venus SW

Project:	Universal Database Library	HAMILTON Bonaduz AG	2010-01-22	Site 8 of 15
Title:	Technical Documentation	Doc. Name:	SRS_UniversalDatabaseLibrary.doc	Version: 0.1

3 hsl-library

3.1 Supported DBMS

Several DBMS are to be supported by hsl-library "Universal_SQL_Database_Library.hsl":

- MS Access
- MS SQL Server
- Oracle
- PostgreSQL
- MySQL

3.2 Connection to DBMS (Drivers / Providers)

To connect from the client operating system (Windows XP) to the specific DBMS on the server, a driver is needed. The driver is depending on the target DBMS and has to be installed on the client computer.

Used drivers / providers:

DBMS	Driver / Provider	Installation
MS Access	Provider=Microsoft.Jet.OLEDB.4.0; Jet OLEDB:Engine Type=5	Should be installed anyway on XP
MS SQL Server	Provider=SQLOLEDB	Should be installed anyway on XP
Oracle	Provider=msdaora	Installation and connecting via service-name. Install needed Network-Config.
PostgreSQL	Driver=PostgreSQL UNICODE	Install driver (possibly modify connection string)
MySQL	Driver=MySQL ODBC 5.1 Driver	Install driver (possibly modify connection string)

ODBC Drivers which are installed on your system: Control Panel --> Administrative Tools --> Data Sources (ODBC) --> Drivers

Project:	Universal Database Library	HAMILTON Bonaduz AG	2010-01-22	Site 9 of 15
Title:	Technical Documentation	Doc. Name:	SRS_UniversalDatabaseLibrary.doc	Version: 0.1

3.3 Functions of hsl-library

3.3.1 Library_Init

Library_Init	<p>Description: Initialize / Create library, objects, and semaphores. Output depending on DBMS.</p> <p>Input-Parameters:</p> <ol style="list-style-type: none"> 1. DB Server Type [STR] 2. DB Name [STR] 3. DB Host Name [STR] 4. DB User Name [STR] 5. DB Password [STR] 6. Trace level for library traces [INT] 7. DB Max. records for SELECT (0 = unlimited, >= 1 Limit) <p>Output-Parameters:</p> <ol style="list-style-type: none"> 1. Delimiter for using keywords as a string [STR] <ul style="list-style-type: none"> • MSAccess: "" • MS SQL: "" • Oracle: "" • PostgreSQL: "" • MySQL: "" 2. Delimiter for field-begin [STR] <ul style="list-style-type: none"> • MSAccess: "[" • MS SQL: "[" • Oracle: "" • PostgreSQL: "\"" • MySQL: "" 3. Delimiter for field-end [STR] <ul style="list-style-type: none"> • MSAccess: "]" • MS SQL: "]" • Oracle: "" • PostgreSQL: "\"" • MySQL: "" 4. Keyword for actual-date-time (of system) [STR] <ul style="list-style-type: none"> • MSAccess: NOW() • MS SQL: GETDATE() • Oracle: sysdate • PostgreSQL: now() • MySQL: NOW() <p>Return: hslTrue = no error, function successful hslFalse = Fatal error, function failed</p>
---------------------	--

Project:	Universal Database Library	HAMILTON Bonaduz AG	2010-01-22	Site 10 of 15
Title:	Technical Documentation	Doc. Name:	SRS_UniversalDatabaseLibrary.doc	Version: 0.1

3.3.2 Library_Delnit

Library_Delnit	Description: De-Initialize / Release library, objects, and semaphores. Input-Parameters: - Output-Parameters: - Return: hslTrue = no error, function successful hslFalse = Fatal error, function failed
----------------	---

3.3.3 DBConnectionOpen

DBConnectionOpen	Description: Open the connection to the DB. Input-Parameters: - Output-Parameters: - Return: hslTrue = no error, function successful hslFalse = Fatal error, function failed
------------------	--

3.3.4 DBConnectionClose

DBConnectionClose	Description: Closes the connection to the DB. Input-Parameters: - Output-Parameters: - Return: hslTrue = no error, function successful hslFalse = Fatal error, function failed
-------------------	--

3.3.5 TransactionBEGIN

TransactionBEGIN	Description: Marks the start of an atomic transaction. Input-Parameters: - Output-Parameters: - Return: hslTrue = no error, function successful hslFalse = Fatal error, function failed
------------------	---

Project:	Universal Database Library	HAMILTON Bonaduz AG	2010-01-22	Site 11 of 15
Title:	Technical Documentation	Doc. Name:	SRS_UniversalDatabaseLibrary.doc	Version: 0.1

3.3.6 TransactionCOMMIT

TransactionCOMMIT	Description: Commits / Apply all changes on DB since previous transaction begin. Input-Parameters: - Output-Parameters: - Return: hslTrue = no error, function successful hslFalse = Fatal error, function failed
--------------------------	---

3.3.7 TransactionROLLBACK

TransactionROLLBACK	Description: Rollbacks / Discards all changes on DB since previous transaction begin. Input-Parameters: - Output-Parameters: - Return: hslTrue = no error, function successful hslFalse = Fatal error, function failed
----------------------------	--

3.3.8 ExecuteSELECT

ExecuteSELECT (including private functions: ExecuteSELECT_Read, ExecuteSELECT_Field)	Description: Executes a select-statement and returns arrays with values for the selected fields. CAUTION: DB Null is converted to "empty string". Input-Parameters: 1. Full SQL-command for select-statement (including field-names) [STR] 2. List with field-names (depending on select) [ARR] Output-Parameters: 1. Number of found records [INT] 2. Array with all found fields. Order depending on SELECT-Statement [ARR] 3. Array: indicating, if field is originally NULL on database. entry for every found field [ARR] Return: hslTrue = no error, function successful hslFalse = Fatal error, function failed
--	--

Project:	Universal Database Library	HAMILTON Bonaduz AG	2010-01-22	Site 12 of 15
Title:	Technical Documentation	Doc. Name:	SRS_UniversalDatabaseLibrary.doc	Version: 0.1

3.3.9 GetFieldFromArrayOfSELECT

GetFieldFromArrayOfSELECT	<p>Description: Gets the needed field out of the array and lists it in an other array. CAUTION: DB Null is "empty string".</p> <p>Input-Parameters:</p> <ol style="list-style-type: none"> 1. Array with fields from select (original from SELECT-Statement) [ARR] 2. Number of fields in input-array [INT] 3. Number of first index of field in array (zero-based) [INT] <p>Output-Parameters:</p> <ol style="list-style-type: none"> 1. Array with values of the needed field (depending on index-number of first field) [ARR] <p>Return: hslTrue = no error, function successful hslFalse = Fatal error, function failed</p>
----------------------------------	---

3.3.10 GetFieldFromArrayOfSELECT_Record

GetFieldFromArrayOfSELECT_Record	<p>Description: Gets the needed field out of the array (of the selected values) and saves it in an variable. CAUTION: DB Null is "empty string".</p> <p>Input-Parameters:</p> <ol style="list-style-type: none"> 1. Array with fields from select (original from SELECT-Statement) [ARR] 2. The record data returned by SELECT [ARR] 3. Index of the required record [INT] 4. Name of field, required column [STR] <p>Output-Parameters:</p> <ol style="list-style-type: none"> 1. Values in the required field in required record <p>Return: hslTrue = no error, function successful hslFalse = Fatal error, function failed</p>
---	---

3.3.11 ExecuteUPDATE

ExecuteUPDATE	<p>Description: Executes a SQL update statement.</p> <p>Input-Parameters: SQL-command for update statement [STR]</p> <p>Output-Parameters: Number of affected records [INT]</p> <p>Return: hslTrue = no error, function successful hslFalse = Fatal error, function failed</p>
----------------------	--

Project:	Universal Database Library	HAMILTON Bonaduz AG	2010-01-22	Site 13 of 15
Title:	Technical Documentation	Doc. Name:	SRS_UniversalDatabaseLibrary.doc	Version: 0.1

3.3.12 ExecuteDELETE

ExecuteDELETE	<p>Description: Executes a SQL delete statement.</p> <p>Input-Parameters: SQL-command for delete statement [STR]</p> <p>Output-Parameters: Number of affected records [INT]</p> <p>Return: hslTrue = no error, function successful hslFalse = Fatal error, function failed</p>
----------------------	--

3.3.13 ExecuteINSERT_OneRecord_ID

ExecuteINSERT_OneRecord_ID	<p>Description: Inserting / add ONE record into a table. Get back ID-value of last inserted record converted to integer. No possible output of NULL-values! Depending on DBMS: MSAccess: SELECT @@IDENTITY MSSQL: SELECT SCOPE_IDENTITY() Oracle: SELECT "Sequence-Name".CURRVAL from DUAL PostgreSQL: SELECT CURRVAL(" Sequence-Name ") MySQL: SELECT LAST_INSERT_ID()</p> <p>Input-Parameters: 1. SQL-command for insert-statement [STR] 2. For Oracle and PostgreSQL: Name of sequences (for table) to query the values of last inserted record [STR]. Other DBMS: empty string "".</p> <p>Output-Parameters: 1. ID of last inserted record [INT]</p> <p>Return: hslTrue = no error, function successful hslFalse = Fatal error, function failed</p>
-----------------------------------	---

3.3.14 ExecuteSTATEMENT

ExecuteSTATEMENT	<p>Description: Executing various SQL-statement such as: Create / Drop View, Inserting more than one record, etc.</p> <p>Input-Parameters: SQL-statement to execute [STR]</p> <p>Output-Parameters: Number of affected records by the statement [INT] -1 = Statements which are not working with records, such as "Drop, Create, etc."</p> <p>Return: hslTrue = no error, function successful hslFalse = Fatal error, function failed</p>
-------------------------	---

Project:	Universal Database Library	HAMILTON Bonaduz AG	2010-01-22	Site 14 of 15
Title:	Technical Documentation	Doc. Name:	SRS_UniversalDatabaseLibrary.doc	Version: 0.1

3.4 Data-Types

Needed comparisons to get the data-type inclusive DB-Null (Standard Venus SW).

.hsl :

IsDBNull (value)	<p>Returns an indication whether the specified variable is of type null (VT_NULL).</p> <p>Parameter value [in] A variable.</p> <p>Return Nonzero if the specified variable is of type null (VT_NULL), otherwise zero.</p>
-----------------------------	---

GetType (var)	<p>The GetType function retrieves the type of the value of a variable.</p> <p>Parameter</p> <p>var A reference to a variable (integer, float or string).</p> <p>Return</p> <p>One of the following string valued constants that indicates the type of the value of a variable:</p> <table><tr><th>Name</th><th>Value</th><th>Meaning</th></tr><tr><td>hslInteger</td><td>"i"</td><td>The value is an integer</td></tr><tr><td>hslFloat</td><td>"f"</td><td>The value is a float</td></tr><tr><td>hslString</td><td>"s"</td><td>The value is a string</td></tr><tr><td></td><td>""</td><td>The value has no type</td></tr></table>	Name	Value	Meaning	hslInteger	"i"	The value is an integer	hslFloat	"f"	The value is a float	hslString	"s"	The value is a string		""	The value has no type
Name	Value	Meaning														
hslInteger	"i"	The value is an integer														
hslFloat	"f"	The value is a float														
hslString	"s"	The value is a string														
	""	The value has no type														

4 Error handling

Basically, errors are handled as far as reasonable.

If no error in the function occurs, the return is TRUE.

If an error with the database or an object occurs, then it's fatal and the error handler is called. The return is FALSE.

5 Concept of wrapper-library

For project specific use of the hsl-library, it's common and well-proved to build a submethod library (.smt) to wrap the base hsl-library. In this wrapper, there are specific functions for the project.

Here the actions to implement such a wrapper-library:

