

SIMATIC IT Unilab 6.7

COM Wrapper

Concepts and User Manual

Preface	
Table of Contents	
Introduction	1
Setting up Com Wrapper	2
Unilab Interface	4
Object Properties	5
Return Codes and Error Numbers	5

Guidelines

This manual contains notices intended to protect the products and connected equipment against damage. These notices are graded according to severity by the following texts:

Caution

Indicates that if the proper precautions are not taken, this can result into property damage.

Notice

Draws your attention to particularly important information on handling the product, the product itself or to a particular part of the documentation.

Trademarks

All names identified by ® are registered trademarks of the Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Preface

Where is this manual valid?

This manual is valid for release 6.7 of SIMATIC IT Unilab.

Basic knowledge required

This guide is intended for SIMATIC IT Unilab users who are responsible for system configuration, such as application managers and system integrators (consultants). To be able to understand the concepts and examples discussed in this guide, the reader should at least have taken the SIMATIC IT Unilab Basic Training.

Purpose

This Concepts and User Manual explains how the Unilab COM Wrapper works.

Readme

The installation includes a readme file, which contains information on upgrade procedures and compatibility with previous releases. This file is supplied both in standard text (**Readme.wri**) and in Acrobat PDF (**Readme.pdf**) format.

Related documentation

The document **Unilab-Interspec Interface** contains information related to the content of this Concepts and User Manual.

For information in merit, please consult the Siemens TSS Website (www.siemens.com/mes-simaticit/tss).

Conventions

The table below describes the specific typographic conventions that are used throughout this manual:

Symbol/Convention	Indicates...
E.g.	Where examples are given.
Text in bold	The names of menus, commands, dialog boxes and toolbar buttons and, in general, all strings (e.g. File menu; Save command).
KEY1+KEY2	Shortcut keys, which permit rapid access to commands (e.g. CTRL+C).
UPPERCASE	The names of keyboard keys (e.g. RETURN key).

Symbol/Convention	Indicates...
<i>Italics</i>	Noun with special importance or significance for which emphasis is needed. The names of parameters that must be replaced with a specific name or value.
>	A succession of commands in which the command preceding the symbol must be selected before the command following it.
Code example	Code example.

SIMATIC IT Documentation Library

The SIMATIC IT Documentation Library provides you with a comprehensive and user-friendly interface to access the overall product documentation where manuals and helps online can be browsed by functionality or by component.

SIMATIC IT Training Center

Siemens IA AS MES offers a number of training courses to familiarize you with the SIMATIC IT product suite. To successfully achieve this goal, training consists of lessons in both theory and practice.

Courses are held year-round, according to a program that is published well in advance of the first scheduled session.

The material on the basis of which our courses are conducted reflects the result of years of experience in process, LIMS, quality control and production management.

All courses are held by expert personnel that are aware of the developments and innovations in the Siemens IA AS MES product suite.

Courses are held in English at the Siemens IA AS MES Training Centers.

Upon request, training courses can also be organized on the customer's premises.

For more information on the training course calendar, please visit our technical web site (<http://www.siemens.com/simatic-it/training>).

SIMATIC IT Service & Support

A comprehensive Software Maintenance program is available with SIMATIC IT products. Software Maintenance includes the following services:

- **Software Update Service (SUS):** automatic distribution of upgrades and service packs
- **Technical Support Service (TSS):** support on technical problems with SIMATIC IT software (standard support and other optional services)
- **Online Support:** a technical web site, providing information such as Frequently Asked Questions and technical documentation on SIMATIC IT products

Software Update Service (SUS)

This service provides automatic shipment of new versions and service packs when released. When a new version / service pack is available for shipping, it is typically shipped within one month.

One copy of the installation DVD is shipped for each Server covered by Software Maintenance.

Hot fixes (officially tested and released) are not shipped and must be downloaded from the Technical Support Service web site.

Technical Support Service (TSS)

Siemens provides a dedicated technical support team for SIMATIC IT products.

The following options are available:

Bronze support: 9 hours/day, 5 days/week

Silver support: 24 hours/day, 5 days/week

Gold support: 24 hours/day, 7 days/week

The principal language of the SIMATIC IT hotline is English.

SIMATIC IT partners and customers covered by the Software Maintenance program are entitled to direct access to the TSS.

Access to TSS

To be able to access TSS, the customer needs to register as a user on the Technical Support web site. Connect to <http://www.siemens.com/mes-simaticit/> and follow the **Technical Support Service** link

The registration form must be completed with:

- Personal data
- The required company and plant information
- The Contract Number provided by Siemens Back Office when the contract is agreed.

Online Support

A customer who is a registered TSS user, can access the Technical Support web site (<http://www.siemens.com/mes-simaticit/tss>), which contains technical information such as:

- Service conditions (Phone numbers, Working hours, Reaction times,...)
- SIMATIC IT knowledge base: a technical support database that includes practical service solutions from Technical Support or the SIMATIC IT community
- SIMATIC IT software (e.g. hot fixes, software examples) and release notes that can be downloaded

- SIMATIC IT cross-industry libraries that can be downloaded (limited access to SIMATIC IT certified partners)
- SIMATIC IT product documentation that can be downloaded
- Frequently Asked Questions and useful tips.

Table of Contents

1	Introduction.....	1-1
1.1	Conceptual Overview.....	1-1
1.2	Architectural Overview.....	1-3
2	Setting up Com Wrapper	2-1
2.1	Oracle Client Installation.....	2-1
3	Unilab Interface.....	3-1
3.1	DB Connection and Transaction Methods.....	3-2
3.1.1	Connect with the currently logged-on Windows or Simatic Logon user.	3-5
3.1.1.1	Handling of Simatic Logon user changes	3-5
3.2	Create Methods	3-6
3.2.1	CreateRequest.....	3-6
3.2.2	CreateSample	3-6
3.2.3	CreateSample2	3-7
3.2.4	CreateNew<Object>Version Methods	3-8
3.3	Select<Object> Methods	3-9
3.4	Cancel<Object> Methods	3-9
3.5	Reanalyse<Object> Methods	3-10
3.6	Add/Remove Methods	3-11
3.6.1	Add<Object> Methods	3-11
3.6.2	Add<yy>To<xx> Methods	3-12
3.6.3	Remove<yy>From<xx> Methods.....	3-14
3.7	Get/Set Methods.....	3-15
3.7.1	Get/Set<Object>Status Methods	3-16
3.7.2	Get/Set1<Object>Property Methods.....	3-17
3.7.3	Get/Set1<xx><yy>Property Methods.....	3-21
3.7.4	Get/Set1<Object>Attribute Methods.....	3-23
3.7.5	Get/Set1<xx><yy>Attribute Methods.....	3-25
3.7.6	Get/Set1<Object>GroupKey Methods	3-27
3.7.7	Get/Set1<Object>Result Methods	3-28
3.7.8	Get/Set1<Object>Spec Methods	3-30
3.7.9	GetAll<Object>Results Methods.....	3-30
3.7.10	Get<Object>List Methods	3-31
3.7.11	Get<xx><yy>List Methods	3-32
3.7.12	GetRqSample method	3-33
3.8	Event Methods.....	3-34
3.8.1	Enabling/Disabling Events	3-34
3.8.2	EventOccurred Event.....	3-34
3.8.3	EventOccurredExt Event.....	3-35
3.9	User Management Methods	3-35
4	Object Properties.....	4-1
4.1	Request Properties	4-1
4.2	Sample Properties	4-1
4.3	ParameterGroup Properties.....	4-2
4.4	Parameter Properties.....	4-3
4.5	Method Properties	4-4
4.6	Cell Properties	4-5
4.7	Request InfoCard Properties	4-5
4.8	Request InfoField Properties	4-6
4.9	Sample InfoCard Properties	4-6
4.10	Sample InfoField Properties	4-6

5	Return Codes and Error Numbers	5-1
5.1	Return Codes.....	5-1
5.2	Error Numbers	5-1

1 Introduction

1.1 Conceptual Overview

SIMATIC IT Unilab is part of the SIMATIC IT suite. Therefore, SIMATIC IT Unilab must provide an easy and straightforward interface for integration with other components and, if applicable, other external products. The **SIMATIC IT Unilab COM Wrapper** meets this requirement.

The **SIMATIC IT Unilab COM Wrapper** is a COM object that offers a very flexible and simple set of methods and events to communicate with SIMATIC IT Unilab. This object is provided by the ActiveX component **UnilabObjectLibrary.dll** (Unilab 6.7 – Object Library). The class name of the object is simply **Unilab**.

The primary focus is on using this COM object to integrate the Unilab environment into the SIMATIC IT framework (and, more specifically, Production Modeler (PM) component. Of course, this COM object can be used in any other application to interact with the Unilab system.

This Unilab object provides:

- Methods offering DB connection and transaction management functionality on the SIMATIC IT Unilab DB
 - **ConnectToDb**
 - **ReturnConnectToDb**
 - **DisconnectFromDB**
 - **SetCustomConnectionParameter**
 - **BeginTransaction**
 - **EndTransaction**
- Methods offering functionality to retrieve data from, send data to or execute commands in SIMATIC IT Unilab

Method	rq	sc	pg	pa	me	ce	scic/rqic	scii/rqii
Create<Object>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>						
Get/Set<Object>Status	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	
Cancel<Object>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	
Reanalyse<Object>			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
Add<Object>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Get/Set1<Object>Property	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Get/Set1<Object>Attribute	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	
Get/Set1<Object>GroupKey	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			<input checked="" type="checkbox"/>			
Get/Set1<Object>Result			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>

Method	rq	sc	pg	pa	me	ce	scic/rqic	scii/rqii
GetAll<Object>Results					<input checked="" type="checkbox"/>			<input checked="" type="checkbox"/>
Get<Object>List	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

- Events allowing notification of certain SIMATIC IT Unilab events in an external system. There are also methods to enable/disable event raising. By default, no events are raised.

- **EventOccurred**
- **EventOccurredExt**
- **EnableEvents**
- **DisableEvents**

The illustration below shows the methods contained in the Unilab class of the **UnilabObjectLibrary.dll** (Unilab 6.7 – Object Library):

Members of 'Unilab'			
➤ AddInfoCardToRequest	➤ CreateNewStVersion	➤ Get1PpAttribute	➤ Get1StPpProperty
➤ AddInfoCardToSample	➤ CreateRequest	➤ Get1PpPrAttribute	➤ Get1StProperty
➤ AddInfoFieldToInfoProfile	➤ CreateSample	➤ Get1PpProperty	➤ Get1UpProperty
➤ AddInfoProfileToRequesttype	➤ CreateUser	➤ Get1PpPrProperty	➤ Get1UsProperty
➤ AddInfoProfileToSampletype	➤ DeleteUser	➤ Get1PrAttribute	➤ GetAllPaResults
➤ AddMethod	➤ DeleteUserProfile	➤ Get1PrMtAttribute	➤ GetAllRqliResults
➤ AddMethodToParameter	➤ DisableEvents	➤ Get1PrMtProperty	➤ GetAllSciiResults
➤ AddParameter	➤ DisableLogging	➤ Get1PrProperty	➤ GetInfoFieldDefStatus
➤ AddParameterGroup	➤ DisconnectFromDB	➤ Get1RqAttribute	➤ GetInfoProfileStatus
➤ AddParameterprofileToRequesttype	➤ EnableEvents	➤ Get1RqGroupKey	➤ GetIpleList
➤ AddParameterprofileToSampletype	➤ EnableLogging	➤ Get1RqlcAttribute	➤ GetMethodDefStatus
➤ AddParameterToParameterprofile	➤ EndTransaction	➤ Get1RqlcProperty	➤ GetMethodStatus
➤ AddSample	➤ EventOccurred	➤ Get1RqliProperty	➤ GetMtCellList
➤ AddSampletypeToRequesttype	➤ EventOccurredExt	➤ Get1RqliResult	➤ GetPaList
➤ BeginTransaction	➤ Get1CeProperty	➤ Get1RqProperty	➤ GetParameterDefStatus
➤ CancelInfoFieldDef	➤ Get1CeResult	➤ Get1RtAttribute	➤ GetParameterGroupStatus
➤ CancelInfoProfile	➤ Get1IeAttribute	➤ Get1RtGroupKey	➤ GetParameterProfileStatus
➤ CancelMethod	➤ Get1IeProperty	➤ Get1RtlpAttribute	➤ GetParameterStatus
➤ CancelMethodDef	➤ Get1IpAttribute	➤ Get1RtlpProperty	➤ GetPpPrList
➤ CancelParameter	➤ Get1IpleAttribute	➤ Get1RtPpAttribute	➤ GetPrMtList
➤ CancelParameterDef	➤ Get1IpleProperty	➤ Get1RtPpProperty	➤ GetRequestInfoCardStatus
➤ CancelParameterGroup	➤ Get1IpProperty	➤ Get1RtProperty	➤ GetRequestStatus
➤ CancelParameterProfile	➤ Get1MeAttribute	➤ Get1RtStAttribute	➤ GetRequesttypeStatus
➤ CancelRequest	➤ Get1MeGroupKey	➤ Get1RtStProperty	➤ GetRqliList
➤ CancelRequestInfoCard	➤ Get1MeProperty	➤ Get1ScAttribute	➤ GetRqSample
➤ CancelRequestType	➤ Get1MeResult	➤ Get1ScGroupKey	➤ GetRtlpList
➤ CancelSample	➤ Get1MtAttribute	➤ Get1ScicAttribute	➤ GetRtPpList
➤ CancelSampleInfoCard	➤ Get1MtCellProperty	➤ Get1ScicProperty	➤ GetRtStList
➤ CancelSampleType	➤ Get1MtProperty	➤ Get1SciiProperty	➤ GetSampleInfoCardStatus
➤ ConnectToDB	➤ Get1PaAttribute	➤ Get1SciiResult	➤ GetSampleStatus
➤ CreateNewIeVersion	➤ Get1PaProperty	➤ Get1ScProperty	➤ GetSampletypeStatus
➤ CreateNewIpVersion	➤ Get1PaResult	➤ Get1StAttribute	➤ GetSciiList
➤ CreateNewMtVersion	➤ Get1PaSpec	➤ Get1StGroupkey	➤ GetStIplList
➤ CreateNewPpVersion	➤ Get1PgAttribute	➤ Get1StIpAttribute	➤ GetStPpList
➤ CreateNewPrVersion	➤ Get1PgProperty	➤ Get1StIpProperty	➤ ReanalyseMethod
➤ CreateNewRtVersion	➤ Get1PgResult	➤ Get1StPpAttribute	➤ ReanalyseParameter

ReanalyseParameterGroup	Set1PpProperty	Set1UpProperty
RemoveInfoFieldFromInfoProfile	Set1PpPrProperty	Set1UsProperty
RemoveInfoProfileFromRequesttype	Set1PrAttribute	SetCustomConnectionParameter
RemoveInfoProfileFromSampletype	Set1PrMtAttribute	SetInfoFieldDefStatus
RemoveMethodFromParameter	Set1PrMtProperty	SetInfoProfileStatus
RemoveParameterFromParameterprofile	Set1PrProperty	SetMethodDefStatus
RemoveParameterProfileFromRequesttype	Set1RqAttribute	SetMethodStatus
RemoveParameterProfileFromSampletype	Set1RqGroupKey	SetParameterDefStatus
RemoveSampletypeFromRequesttype	Set1RqlcAttribute	SetParameterGroupStatus
ReturnConnectToDB	Set1RqlcProperty	SetParameterProfileStatus
SelectRequest	Set1RqliProperty	SetParameterStatus
SelectSample	Set1RqliResult	SetRequestInfoCardStatus
Set1CeProperty	Set1RqProperty	SetRequestStatus
Set1CeResult	Set1RtAttribute	SetRequestTypeStatus
Set1IeAttribute	Set1RtGroupkey	SetSampleInfoCardStatus
Set1IeProperty	Set1RtIpAttribute	SetSampleStatus
Set1IpAttribute	Set1RtIpProperty	SetSampleTypeStatus
Set1IpleAttribute	Set1RtPpAttribute	TokenEncrypt
Set1IpleProperty	Set1RtPpProperty	
Set1IpProperty	Set1RtProperty	
Set1MeAttribute	Set1RtStAttribute	
Set1MeGroupKey	Set1RtStProperty	
Set1MeProperty	Set1ScAttribute	
Set1MeResult	Set1ScGroupKey	
Set1MtAttribute	Set1ScIcAttribute	
Set1MtCellProperty	Set1ScIcProperty	
Set1MtProperty	Set1ScIiProperty	
Set1PaAttribute	Set1ScIiResult	
Set1PaProperty	Set1ScProperty	
Set1PaResult	Set1StAttribute	
Set1PaSpec	Set1StGroupkey	
Set1PgAttribute	Set1StIpAttribute	
Set1PgProperty	Set1StIpProperty	
Set1PgResult	Set1StPpAttribute	
Set1PpAttribute	Set1StPpProperty	
Set1PpPrAttribute	Set1StProperty	

1.2 Architectural Overview

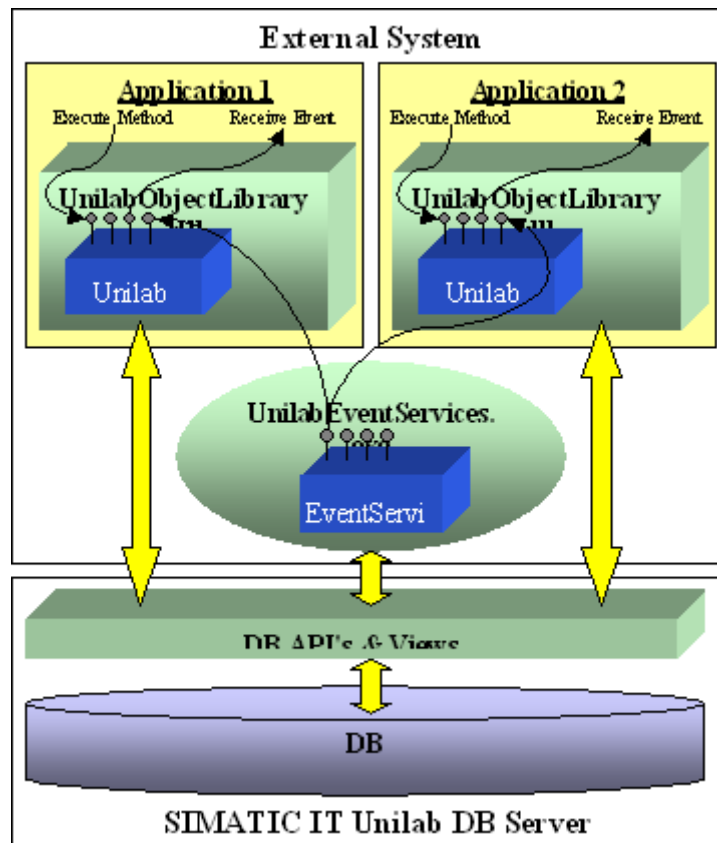
Each external application that wants to communicate with SIMATIC IT Unilab through the **Unilab COM Wrapper** needs to have a reference to an instance of the Unilab class.

Each instance of the Unilab class has its own connection to the SIMATIC IT Unilab DB.

In order to pass events from Unilab to Production Modeler in an efficient manner, a COM server named **UnilabEventServices.exe** is installed for each Unilab database. No matter how many **Unilab** instances have enabled event-handling, there is only one unique **EventService** instance (identified by a unique Service Name). An **EventService** instance dispatches the events detected on the database to one or more interested **Unilab** instances. An EventService instance has its own connection to the SIMATIC IT Unilab DB.

On the database, access to the tables is provided through APIs & Views.

The illustration below shows the architectural overview of the Unilab interface.



2 Setting up Com Wrapper

In order to use Com Wrapper you must do the following:

- Install Oracle Client
- Execute RegisterComWrapper.bat script with **administrator** rights (this can be found at the following location: *<InstallDir>\SIT\Unilab 6.7\COM Wrapper\RegisterComWrapper.bat*)

2.1 Oracle Client Installation

Important

In order to execute the PowerShell script, the execution of scripts has to be enabled on the system.

To enable script execution, execute the following command in PowerShell:

Set-ExecutionPolicy RemoteSigned

Answer with Y(es) when prompted by the system.

Make sure the PowerShell script is executed with **admin** rights!

The installation package does not include Oracle client. The Oracle client files must be included after installation.

Unilab 6.7 is using ODP.NET 4.11.2.0.2 as part of ODAC 11.2.0.2 or Oracle full client 11.2.0.2.

If there is no Oracle 11.2.0.2 installation present on your system, download ODAC from <http://www.oracle.com/technetwork/database/windows/downloads/utilsoft-087491.html>.

Note

In order to download the ODAC package, a free Oracle Web Account is needed.

To facilitate installation on multiple systems, you can store the downloaded and extracted package on a network share. Once downloaded and extracted, do the following:

- Start PowerShell console as an administrator;
- Execute the Run-CopyOracleFiles.ps1 found in the *<InstallDir>\SIT\Siemens\OracleInstantClient* folder (e.g.: *c:\Siemens\SIT\Unilab 6.7\OracleInstantClient*);
- When prompted, specify the following:
 - The source location of the Oracle files.

Note: This can be the path to the downloaded and extracted package (local or network share). You can omit this if you want to use the Oracle files from an existing installation.

- The destination path for the Oracle files.

Note: By default, this is the same location as the script, therefore specifying it is optional.

3 Unilab Interface

Every method that accesses one of the SIMATIC IT Unilab objects has to identify that object by its **<Object Key>**. This **<Object Key>** is represented by one or more parameters of the method.

The tables below list the acronyms and abbreviations that are used throughout this manual:

Object Keys for configuration objects

<xx>/object	<Object Key>
ie/infocfielddefinition	<ie><version>
pp/parameterprofile	<pp><version><pp_key1...5>
pr/parameterdefinition	<pr><version>
st/sampletype	<st><version>
ip/infoprofile	<ip><version>
rt/requesttype	<rt><version>
mt/methoddefinition	<mt><version>

Object Keys for operational objects

<xx>/object	<Object Key>
rq/request	<rq>
sc/sample	<sc>
pg/parametergroup	<sc>,<pg>
pa/parameter	<sc>,<pg>,<pa>
me/method	<sc>,<pg>,<pa>,<me>
ce/cell	<sc>,<pg>,<pa>,<me>,<ce>
rqic/requestinfocard	<rq>,<ic>
rqii/requestinfofield	<rq>,<ic>,<ii>
scic/sampleinfocard	<sc>,<ic>
scii/sampleinfofield	<sc>,<ic>,<ii>

Example

Here is an example of how you can access a method through an object key.

```
Get1<xx>Property (<Object Key>, PropertyName,  
PropertyValue)  
↓  
Get1PaProperty (sc, pg, pa, PropertyName, PropertyValue)
```

Tip

For an overview of the possible return codes of a method or the raised error numbers and descriptions, see Return Codes and Error Numbers.

Method types

The **Unilab** interface handles the following method types:

- DB Connection and Transaction Methods
- Create<Object> Methods
- Create Methods
- Select<Object> Methods
- Cancel<Object> Methods
- Reanalyse<Object> Methods
- Add/Remove Methods
- Get/Set Methods
- Event Methods
- User Management Methods

Methods with restrictions concerning input-argument values

A number of methods provided with the DB API present limitations as to the values that their input arguments may assume. Therefore, when entering these arguments for any DB API method, we strongly suggest consulting the DB API online help for indications as to supported types.

For example:

DB API method **CreateRequest**

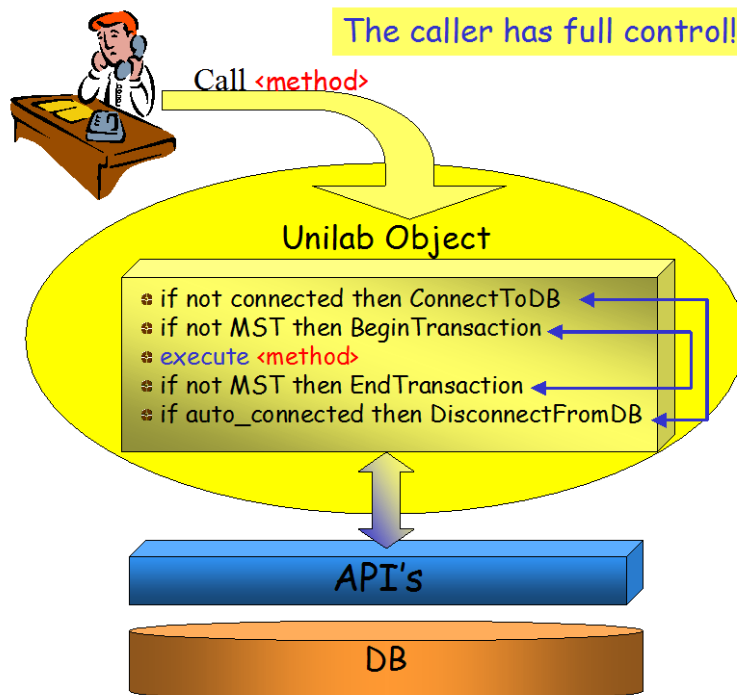
a_field_types (IN)

Array containing the type of the field to pass. The supported types are: **gk**, **rq**.

3.1 DB Connection and Transaction Methods

By default, each method can be called without specifying a DB name and user/password. In this case, each method is automatically treated as a single transaction.

The figure below illustrates DB connection and transaction management.



However, if necessary, the caller can take full control over DB connection and transaction management, using the following methods: **ConnectToDb**, **ReturnConnectToDb**, **DisConnectFromDb**, **SetCustomConnectionParameter**, **BeginTransaction** and **EndTransaction**.

Important

If the caller wants to take control, he must keep the COM object instance alive between consecutive calls, in order to preserve the context.

ConnectToDB

Connects to the Unilab DB.

If not specified, the default DbName, UserName and Password – as configured in the registry¹ – will be used. Any pending transaction will be committed. Any open DB connection will be closed.

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
DbName	String	ByVal	Yes (")
UserName	String	ByVal	Yes (")
Password	String	ByVal	Yes (")
ChangePermanently	Boolean	ByVal	Yes (False)

ReturnConnectToDB

Returns the database object created by ConnectToDB.

¹ Please refer to the Simatic IT Unilab – Registry settings document for a detailed description on the registry settings used.

As with ConnectToDb, if not specified, the default Database, User and PassWD will be used. Any pending transaction will be committed. Any open DB connection will be closed.

The COM-wrapper does not offer (yet) the full coverage of all business logic contained in DB-API calls. This method allows the integration of any DB-API call into the same transaction logic with regular COM-wrapper call. An example project on how to do this is given in the COM-wrapper folder (on the SIMATIC IT Unilab installation folder).

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
extobjDatabase	OraDatabase object	ByRef	No
Database	String	ByVal	Yes ("")
User	String	ByVal	Yes ("")
PassWD	String	ByVal	Yes ("")
ChangePerm	Boolean	ByVal	Yes (False)

DisconnectFromDB

Disconnects from the Unilab DB.

Any pending transaction will be committed.

Parameters: None.

SetCustomConnectionParameter

This function is called by all standard Unilab client applications and background jobs. The client applications pass the value contained in the Unilab registry setting **ConnectionCustomParameter** to that API. That API function will then call the customizable package function **uncustomsetconnection.SetCustomConnectionParameter** with the specified argument value.

The default implementation allows turning on the Oracle tracing (please refer to the Simatic IT Unilab – Registry settings document, as well as all possible logging for details).

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
customConnectionParameter	String	ByVal	No

BeginTransaction

Marks the beginning of a MST.

Transactions cannot be nested. If the function is called again before ending the previous transaction, a warning code is returned (without altering the pending transaction status).

Parameters: None

EndTransaction

Ends and commits or rolls back the current MST.

If there is no MST in progress, a warning code is returned.

Name	Type	ByVal/ByRef	Optional (Default Value)
Commit	Boolean	ByVal	Yes (True)

3.1.1 Connect with the currently logged-on Windows or Simatic Logon user.

With the COM wrapper\ConnectionSystem registry setting set to **Windows** or **SLS**, both:

- The implicit connection (by executing a method without executing a ConnectToDB call)
- The explicit connection (by executing explicitly the ConnectToDB call)

will use respectively the currently logged-on Windows or Simatic Logon user to build up the database connection. The COM wrapper\ConnectionString setting defines the layout of the connection string that will be applied.²

3.1.1.1 Handling of Simatic Logon user changes

There are 2 options to handle a user change of the Simatic Logon user.

- If working disconnected (KeepDBConnection = 0), the Simatic Logon user change will be handled automatically.
- If working with a permanent connection, it is up to the application that uses the COM –wrapper to monitor the Simatic Logon User Changes, and to implement the appropriate reaction to the User Change event. How this can be done, is described in the Simatic Logon – Programming guide (by default, in the C:\Program Files\Siemens\SIMATICLogon\developmentkit folder).

² Refer to the manual Simatic IT Unilab – Security – Database and Logon for more details regarding the different authentication mechanisms, and the supported combinations.

3.2 Create Methods

The **Unilab** interface can handle the following Create methods:

- CreateRequest
- CreateSample
- CreateSample2
- CreateNew<Object>Version Methods

3.2.1 CreateRequest

Creates a request of a specific type.

The request code that uniquely identifies the request can be passed as a parameter. However, there is always the risk that this code is not unique in the DB. In this case, Unilab will create the request, but generate its own unique request code (as occurs by default in case the RequestCode parameter was left empty).

It is possible to specify the context in the a_field_xxx arguments.

The a_field_types, a_field_names and a_field_arguments should be arrays.

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
RequestType	String	ByVal	No
RequestCode	String	ByRef	Yes ("")
Why	String	ByVal	Yes ("")
a_field_types	Variant	ByRef	Yes
a_field_names	Variant	ByRef	Yes
a_field_values	Variant	ByRef	Yes
a_field_nr_of_rows	integer	ByVal	Yes (0)

3.2.2 CreateSample

Creates a sample of a specific type.

The sample code that uniquely identifies the sample can be passed as a parameter. However, there is always the risk that this code is not unique in the DB. In this case, Unilab will create the sample, but generate its own unique sample code (as occurs by default in case the SampleCode parameter was left empty).

It is possible to specify the context in the a_field_xxx arguments.

The a_field_types, a_field_names and a_field_arguments should be arrays.

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
SampleType	String	ByVal	No

Name	Type	ByVal/ByRef	Optional (Default Value)
SampleCode	String	ByRef	Yes ("")
Why	String	ByVal	Yes ("")
a_field_types	Variant	ByRef	Yes
a_field_names	Variant	ByRef	Yes
a_field_values	Variant	ByRef	Yes
a_fField_nr_of_rows	integer	ByVal	Yes (0)

Example:

The following piece of VB-code creates a sample of type 'Waste water' with context: plant scgk= 'Ninove'

```
Dim unilab_object As New Unilab
Dim ret_code As Integer
ret_code = unilab_object.ConnectToDB("ultralix_u50sp3",
"se", "se")
Dim a_field_types(0) As String
Dim a_field_names(0) As String
Dim a_field_values(0) As String
Dim a_field_nr_of_rows As Integer
Dim sc As String
a_field_types(0) = "gk"
a_field_names(0) = "plant"
a_field_values(0) = "Ninove"
a_field_nr_of_rows = 1
ret_code = unilab_object.CreateSample("Waste Water", sc,
"test", a_field_types, a_field_names, a_field_values,
a_field_nr_of_rows)
MsgBox (sc)
```

3.2.3 CreateSample2

Like CreateSample, this method creates a sample of a specific type, but with the possibility of not automatically creating the associated parameter groups and/or info cards.

See method CreateSample for additional details.

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
SampleType	String	ByVal	No
SampleCode	String	ByRef	Yes ("")
CreateIC	String	ByVal	Yes ("")
CreatePG	String	ByVal	Yes ("")
Why	String	ByVal	Yes ("")
a_field_types	Variant	ByRef	Yes
a_field_names	Variant	ByRef	Yes
a_field_values	Variant	ByRef	Yes
a_fField_nr_of_rows	integer	ByVal	Yes (0)

3.2.4 CreateNew<Object>Version Methods

The **CreateNewVersion** method can create a new major or minor version of an existing object. If there does not yet exist an object with the same object_id, then a complete new object with version **UNVERSION.P_INITIAL_VERSION** will be created.

The table below lists the parameters for methods where <Object> is not 'pp':

Name	Type	ByVal/ByRef	Optional (Default Value)
<Object Key>	String	ByVal	No
A_VERSION	String	ByRef	No
Why	String	ByVal	Yes ("")
Previous_version	String	ByVal	Yes ("~Highest~")

The version argument is an in/out argument and can have two special input values:

- ~minor~: this will create a new minor version of the object, this will be used if the argument is empty ("")
- ~major~: this will create a new major version of the object.

The version argument will return the new created version of the object.

CreateNewPpVersion Method

Creates a new version of a parameter profile.

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
ParameterProfile	String	ByVal	No
A_VERSION	String	ByRef	No
Why	String	ByVal	Yes ("")
Previous_version	String	ByVal	Yes ("~Highest~")
Pp_key1	String	ByVal	Yes (" ")
Pp_key2	String	ByVal	Yes (" ")
Pp_key3	String	ByVal	Yes (" ")
Pp_key4	String	ByVal	Yes (" ")
Pp_key5	String	ByVal	Yes (" ")

3.3 Select<Object> Methods

The following <Object> types are supported:

- Request
- Sample

The Select<Object> methods return a list (arrays) of objects.

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
a_col_id	String[]	ByRef	No
a_col_tp	String[]	ByRef	No
a_col_value	String[]	ByRef	No
a_col_nr_of_rows	Integer	ByVal	No
<Object Key>	String[]	ByRef	No
A_DESCRIPTION	String[]	ByRef	No
A_SS	String[]	ByRef	No

Example:

The following piece of VB-code returns the list of requests of the request type 'Food'

```
Dim unilab_object As New Unilab
Dim ret_code As Integer
ret_code = unilab_object.ConnectToDB("ultralix_u50sp3",
"se", "se")
Dim a_col_id(0) As String
Dim a_col_tp(0) As String
Dim a_col_value(0) As String
Dim a_col_nr_of_rows As Integer
Dim rq() As String
Dim a_description() As String
Dim a_ss() As String
a_col_id(0) = "rt"
a_col_tp(0) = "rq"
a_col_value(0) = "Food"
a_col_nr_of_rows = 1
ret_code = unilab_object.SelectRequest(a_col_id, a_col_tp,
a_col_value, a_col_nr_of_rows, rq, a_description, a_ss)
Dim i As Integer
If ret_code = 0 Then
    For i = 0 To UBound(rq)
        MsgBox (rq(i))
    Next
End If
```

3.4 Cancel<Object> Methods

The following <Object> types are supported:

Operational	Configuration
Request	Request type
Sample	Sample type
Parameter	ParameterDef
Parameter Group	ParameterProfile
Method	MethodDef
InfoCard	InfoProfile
	InfoFieldDef

This method will cancel an <Object> and call the **unapiprp.CancelObject** API.

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
<Object Key>	String	ByVal	No
Why	String	ByVal	Yes ("")
Version	String	ByVal	Yes ("~Highest~")

Important

An <Object> can be cancelled only if a “manual” transition from the current status to the cancelled status is authorized in the life cycle. Remember that the identity of the user can be of importance in this “authorization” process.

CancelParameterProfile

Cancels a parameter profile and calls the **unapippp.CancelParameterProfile** API.

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
ParameterProfile	String	ByVal	No
Why	String	ByVal	Yes ("")
Version	String	ByVal	Yes ("~Highest~")
Pp_key1	String	ByVal	Yes (" ")
Pp_key2	String	ByVal	Yes (" ")
Pp_key3	String	ByVal	Yes (" ")
Pp_key4	String	ByVal	Yes (" ")
Pp_key5	String	ByVal	Yes (" ")

3.5 Reanalyse<Object> Methods

The following <Object> types are supported:

- ParameterGroup
- Parameter
- Method.

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
<Object>	String	ByVal	No
Why	String	ByVal	Yes ("")

Important

An <Object> can be reanalyzed only if a “manual” transition from the current status to the reanalyze status is authorized in the life cycle. Remember that the identity of the user can be of importance in this “authorization” process.

3.6 Add/Remove Methods

The **Unilab** interface can handle the following Add/Remove methods:

- Add<Object> Methods
- Add<yy>To<xx> Methods
- Remove<yy>From<xx> Methods

3.6.1 Add<Object> Methods

The following <Object> types are supported:

- Sample
- ParameterGroup
- Parameter
- Method
- InfoCardToSample/InfoCardToRequest.

Add<Object = Sample>

Creates and adds a sample to a request.

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
RequestCode	String	ByVal	No
SampleType	String	ByVal	No
SampleCode	String	ByRef	No
UserName	String	ByVal	No
Why	String	ByVal	Yes ("")

Important

Both the request and the sample type must already exist.

Add<Object = Parameter / Method / InfoCardToSample / InfoCardToRequest>

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
<Object>	String	ByVal	No
Why	String	ByVal	Yes ("")
CreationType	Integer	ByVal	Yes (-2)

The following values for the **CreationType** parameter are supported:

- 2 (MOD_FLAG_INSERT): inserts the <Object>
- 4 (MOD_FLAG_CREATE): inserts the <Object> and creates all details
- 7 (MOD_FLAG_INSERT_AND_CRAU): inserts the <Object> and inherits all attributes.

Important

A new <Object> is created and added to a parent <Object>.

Add<Object = ParameterGroup>

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
SampleCode	String	ByVal	No
ParameterProfile	String	ByVal	No
Why	String	ByVal	Yes ("")
CreationType	Integer	ByVal	Yes (-2)
Pp_key1	string	ByVal	Yes (" ")
Pp_key2	string	ByVal	Yes (" ")
Pp_key3	string	ByVal	Yes (" ")
Pp_key4	string	ByVal	Yes (" ")
Pp_key5	string	ByVal	Yes (" ")

The following values for the **CreationType** parameter are supported:

- 2 (MOD_FLAG_INSERT): inserts the <Object>
- 4 (MOD_FLAG_CREATE): inserts the <Object> and creates all details
- 7 (MOD_FLAG_INSERT_AND_CRAU): inserts the <Object> and inherits all attributes.

Important

A new Parameter group is created and added to a parent sample.

3.6.2 Add<yy>To<xx> Methods

These methods will add an object to a list.

The following Add<yy>To<xx> are supported:

- AddInfoprofileToSampletype
- AddParameterprofileToSampletype

- AddSamplotypeToRequesttype
- AddParameterprofileToRequesttype
- AddInfoprofileToRequesttype
- AddParameterToParameterProfile
- AddMethodToParameter
- AddInfofieldToInfoprofile

The table below lists the parameters for methods where <xx> and <yy> are not 'parameterprofile':

Name	Type	ByVal/ByRef	Optional (Default Value)
<xx>(abbreviation)	String	ByVal	No
<yy> (abbreviation)	String	ByRef	No
Version	String	ByVal	Yes ("~Highest~")
<yy>_version	String	ByVal	Yes ("~Current~")
Why	String	ByVal	Yes ("")

AddInfocardToRequest method

Adds an infocard to a request

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
RequestCode	String	ByVal	No
InfoProfile	String	ByVal	No
Why	String	ByVal	Yes ("")
CreationType	Integer	ByVal	RqInsert

AddInfocardToSample method

Adds an infocard to a sample

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
SampleCode	String	ByVal	No
InfoProfile	String	ByVal	No
Why	String	ByVal	Yes ("")
CreationType	Integer	ByVal	lcInsert

AddParameterProfileTo<xx> methods

Supported <xx> : Samplotype, Requesttype

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
<xx>	String	ByVal	No
pp	String	ByRef	No

Name	Type	ByVal/ByRef	Optional (Default Value)
Version	String	ByVal	Yes ("~Highest~")
pp_version	String	ByVal	Yes ("~Current~")
Why	String	ByVal	Yes ("")
Pp_key1	String	ByVal	Yes (" ")
Pp_key2	String	ByVal	Yes (" ")
Pp_key3	String	ByVal	Yes (" ")
Pp_key4	String	ByVal	Yes (" ")
Pp_key5	String	ByVal	Yes (" ")

AddParameterToParameterprofile method

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
ParameterProfile	String	ByVal	No
Pr	String	ByRef	No
Version	String	ByVal	Yes ("~Highest~")
pr_version	String	ByVal	Yes ("~Current~")
Why	String	ByVal	Yes ("")
Pp_key1	String	ByVal	Yes (" ")
Pp_key2	String	ByVal	Yes (" ")
Pp_key3	String	ByVal	Yes (" ")
Pp_key4	String	ByVal	Yes (" ")
Pp_key5	String	ByVal	Yes (" ")

3.6.3 Remove<yy>From<xx> Methods

Removes a <yy> in the <yy> list of <xx>. The method will remove the first matching <yy> in the list.

The table below lists the parameters for methods where <xx> and <yy> are not 'parameterprofile'.

Name	Type	ByVal/ByRef	Optional (Default Value)
<xx>	String	ByVal	No
<yy> (abbreviation)	String	ByRef	No
Version	String	ByVal	Yes ("~Highest~")
Why	String	ByVal	Yes ("")

RemoveParameterProfileFrom<xx> method

Supported <xx> : Sampletype, Requesttype,.

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
<xx>	String	ByVal	No

Name	Type	ByVal/ByRef	Optional (Default Value)
pp	String	ByRef	No
Version	String	ByVal	Yes ("~Highest~")
Why	String	ByVal	Yes ("")
Pp_key1	String	ByVal	Yes (" ")
Pp_key2	String	ByVal	Yes (" ")
Pp_key3	String	ByVal	Yes (" ")
Pp_key4	String	ByVal	Yes (" ")
Pp_key5	String	ByVal	Yes (" ")

RemoveParameterFromParameterProfile method

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
ParameterProfile	String	ByVal	No
Pr	String	ByRef	No
Why	String	ByVal	Yes ("")
Version	String	ByVal	Yes ("~Highest~")
Pp_key1	String	ByVal	Yes (" ")
Pp_key2	String	ByVal	Yes (" ")
Pp_key3	String	ByVal	Yes (" ")
Pp_key4	String	ByVal	Yes (" ")
Pp_key5	String	ByVal	Yes (" ")

3.7 Get/Set Methods

The **Unilab** interface can handle the following Get/Set methods:

- Get/Set<Object>Status Methods
- Get/Set1<Object>Property Methods
- Get/Set1<xx><yy>Property Methods
- Get/Set1<Object>Attribute Methods
- Get/Set1<xx><yy>Attribute Methods
- Get/Set1<Object>GroupKey Methods
- Get/Set1<Object>Result Methods
- Get/Set1<Object>Spec Methods
- GetAll<Object>Results Methods
- Get<Object>List Methods
- Get<xx><yy>List Methods
- GetRqSample method

3.7.1 Get/Set<Object>Status Methods

The following <Object> types are supported:

Operational	Configuration
Request	Request type
Sample	Sample type
Parameter	ParameterDef
Parameter Group	ParameterProfile
Method	MethodDef
	InfoProfile
	InfoFieldDef

Get<Object>Status

Gets the current status information of an <Object>.

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
<Object Key>	String	ByVal	No
StatusCode	String	ByRef	Yes ("")
StatusDescription	String	ByRef	Yes ("")
Version	String	ByVal	Yes (the default value depends on <Object>)

Set<Object>Status

Changes the status of an <Object>. This method will call the API **UNAPIPRP.ChangeObjectStatus**.

Important

Changing the status of an <Object> can take place only if a 'manual' transition from the current status to the specified new status is authorized in the life cycle. Remember that the identity of the user can be of importance in this "authorization" process.

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
<Object Key>	String	ByVal	No
NewStatusCode	String	ByVal	No
Why	String	ByVal	Yes ("")
Version	String	ByVal	Yes (the default value depends on <Object>)

Important

Changing the status of an <Object> can take place only if a “manual” transition from the current status to the specified new status is authorized in the life cycle. Remember that the identity of the user can be of importance in this “authorization” process.

GetParameterProfileStatus

Gets the current status information of an <Object>.

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
ParameterProfile	String	ByVal	No
StatusCode	String	ByRef	Yes (“”)
StatusDescription	String	ByRef	Yes (“”)
Version	String	ByVal	Yes (“~Current~”)
Pp_key1	String	ByVal	Yes (“ ”)
Pp_key2	String	ByVal	Yes (“ ”)
Pp_key3	String	ByVal	Yes (“ ”)
Pp_key4	String	ByVal	Yes (“ ”)
Pp_key5	String	ByVal	Yes (“ ”)

SetParameterProfileStatus

Changes the status of a parameter profile. This method will call the API **UNAPIPPP.ChangePpStatus**.

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
ParameterProfile	String	ByVal	No
Version	String	ByVal	No
NewStatusCode	String	ByVal	No
Why	String	ByVal	Yes (“”)
Pp_key1	String	ByVal	Yes (“ ”)
Pp_key2	String	ByVal	Yes (“ ”)
Pp_key3	String	ByVal	Yes (“ ”)
Pp_key4	String	ByVal	Yes (“ ”)
Pp_key5	String	ByVal	Yes (“ ”)

3.7.2 Get/Set1<Object>Property Methods

The following <Object> types are supported:

- rq
- sc
- pg
- pa
- st
- rt
- pr
- mt

- me
- ce
- pp
- ic
- ii
- ip
- ie

See also...

For an overview of the supported **PropertyName** values, see Object Properties.

- **Request Properties** (rq)
- **Sample Properties** (sc)
- **ParameterGroup Properties** (pg)
- **Parameter Properties** (pa)
- **Method Properties** (me)
- **Cell Properties** (ce)
- **Request InfoCard Properties** (rqic)
- **Request InfoField Properties** (rqii)
- **Sample InfoCard Properties** (scic)
- **Sample InfoField Properties** (scii)

Get1<Object>Property

Gets the value of the specified property of <Object> with <Object> in st, rt, , pr, mt, ip, ie, rq, sc, pg, pa, me, rqic, scic, rqii, scii, mecell.

The supported properties are the out-arguments of the Get<Object> DB-API and ss_description, lc_description.

Example: the supported properties for Get1StProperty are:

```
(sql: SELECT LOWER(SUBSTR(a.argument_name , 3))
      FROM all_arguments a
      WHERE object_name=UPPER('GetSampleType')
      AND package_name='UNAPIST' AND argument_name IS NOT
NULL
      AND in_out = 'OUT'
      ORDER BY sequence)

st
version
version_is_current
effective_from
effective_till
description
description2
is_template
confirm_userid
shelf_life_val
shelf_life_unit
nr_planned_sc
freq_tp
freq_val
```



```
freq_unit
invert_freq
last_sched
last_cnt
last_val
priority
label_format
descr_doc
descr_doc_version
allow_any_pp
sc_uc
sc_uc_version
sc_lc
sc_lc_version
inherit_au
inherit_gk
st_class
log_hs
allow_modify
active
lc
lc_version
ss
```

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
<Object Key>	String	ByVal	No
PropertyName	String	ByVal	No
PropertyValue	Variant	ByRef	No
Version (only for configuration objects)	String	ByVal	Yes (the default value depends on the Object Key)

There are two special values for the Version argument:

- Version = "~Current~" will look at the current version of object <xx>
- Version = "~Highest~" will look at the highest version of object <xx>

Get1PpProperty

Gets the value of the specified property of a parameter profile.

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
ParameterProfile	String	ByVal	No
PropertyName	String	ByVal	No
PropertyValue	Variant	ByRef	No
Version	String	ByVal	Yes ("~Current~")
Pp_key1	String	ByVal	Yes (" ")
Pp_key2	String	ByVal	Yes (" ")
Pp_key3	String	ByVal	Yes (" ")
Pp_key4	String	ByVal	Yes (" ")

Name	Type	ByVal/ByRef	Optional (Default Value)
Pp_key5	String	ByVal	Yes (" ")

Set1<Object>Property

Sets the value of the specified property of <Object> with <Object> in st, rt, pr, mt, ip, ie, rq, sc, pg, pa, me, rqc, scic, rqii, scii, mecell.

This method cannot be used to change fields that are part of the object key (e.g. version).

The supported properties (except for modify_reason) are the in-arguments of the save<xx> DB-API.

Example: The supported properties of Set1StProperty are:

- st
- version
- version_is_current
- effective_from
- effective_till
- description
- description2
- is_template
- confirm_userid
- shelf_life_val
- shelf_life_unit
- nr_planned_sc
- freq_tp
- freq_val
- freq_unit
- invert_freq
- last_sched
- last_cntlast_cnt
- last_val
- priority
- label_format
- descr_doc
- descr_doc_version
- allow_any_pp
- sc_uc
- sc_uc_version
- sc_lc
- sc_lc_version
- inherit_au
- inherit_gk
- st_class
- log_hs
- lc
- lc_version

Remark: The property list for the get function is not the same as the property list for the set function. 'active' is an allowed property for the get function and not for the set function.

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
<Object Key>	String	ByVal	No
PropertyName	String	ByVal	No
PropertyValue	Variant	ByRef	No
Version (only for configuration objects)	String	ByVal	Yes (the default value depends on the Object Key)
Why	String	ByVal	Yes ("")

Set1PpProperty

Sets the value of the specified property of a parameter profile

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
ParameterProfile	String	ByVal	No
PropertyName	String	ByVal	No
PropertyValue	Variant	ByRef	No
Version	String	ByVal	Yes ("~Highest~")
Pp_key1	String	ByVal	Yes (" ")
Pp_key2	String	ByVal	Yes (" ")
Pp_key3	String	ByVal	Yes (" ")
Pp_key4	String	ByVal	Yes (" ")
Pp_key5	String	ByVal	Yes (" ")

3.7.3 Get/Set1<xx><yy>Property Methods

The following <xx><yy> are supported:

- stip • rtp • pppr • mtcell
- stpp • rtip • prmt • rtst

Get1<xx><yy>Property

Gets the value of the specified property of an <xx><yy>, The method will search for the first occurrence of <yy> in <xx>.

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
<xx>	String	ByVal	No
<yy>	String	ByVal	No
PropertyName	String	ByVal	No
PropertyValue	Variant	ByRef	No
Version (only for configuration objects)	String	ByVal	Yes (the default value depends on the Object Key)

Get1PpPrProperty

Gets the value of the specified property of a parameter profile

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
pp	String	ByVal	No
pr	String	ByVal	No

Name	Type	ByVal/ByRef	Optional (Default Value)
PropertyName	String	ByVal	No
PropertyValue	Variant	ByRef	No
Version	String	ByVal	Yes (“~Current~”)
Pp_key1	String	ByVal	Yes (“ ”)
Pp_key2	String	ByVal	Yes (“ ”)
Pp_key3	String	ByVal	Yes (“ ”)
Pp_key4	String	ByVal	Yes (“ ”)
Pp_key5	String	ByVal	Yes (“ ”)

Set1<xx><yy>Property

Sets the value of the specified property of <xx><yy>. This method cannot be used to change fields that are part of the object key (e.g. version).

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
<xx>	String	ByVal	No
<yy>	String	ByVal	No
PropertyName	String	ByVal	No
PropertyValue	String	ByVal	No
Version (only for configuration objects)	String	ByVal	Yes (the default value depends on the Object Key)
Why	String	ByVal	Yes (“ ”)

Set1<xx>PpProperty

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
<xx>	String	ByVal	No
pp	String	ByVal	No
PropertyName	String	ByVal	No
PropertyValue	String	ByRef	No
Version	String	ByVal	Yes (“~Highest~”)
Why	String	ByVal	Yes (“ ”)
Pp_key1	String	ByVal	Yes (“ ”)
Pp_key2	String	ByVal	Yes (“ ”)
Pp_key3	String	ByVal	Yes (“ ”)
Pp_key4	String	ByVal	Yes (“ ”)
Pp_key5	String	ByVal	Yes (“ ”)

Set1PpPrProperty

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
ParameterProfile	String	ByVal	No
pr	String	ByVal	No
PropertyName	String	ByVal	No
PropertyValue	String	ByRef	No
Version	String	ByVal	Yes ("~Highest~")
Why	String	ByVal	Yes ("")
Pp_key1	String	ByVal	Yes (" ")
Pp_key2	String	ByVal	Yes (" ")
Pp_key3	String	ByVal	Yes (" ")
Pp_key4	String	ByVal	Yes (" ")
Pp_key5	String	ByVal	Yes (" ")

3.7.4 Get/Set1<Object>Attribute Methods

The following <Object> types are supported:

- rq
- sc
- pg
- pa
- me
- scic
- rqic
- rt
- st
- pr
- mt
- ip
- ie

Get1<Object>Attribute

Gets the value of the specified attribute of <Object> with <Object> in the <Object> types given in the list above.

The method will call the API **unapiprp.GetObjectAttribute**.

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
AttributeName	String	ByVal	No
AttributeValue	String	ByRef	No
Version (only for configuration objects)	String	ByVal	Yes ("the default value depends on the Object Key")

Get1PpAttribute

Gets the value of the specified attribute of a parameter profile.

The method will call the API **unapiprp.GetParameterProfileAttribute**.

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
ParameterProfile	String	ByVal	No
AttributeName	String	ByVal	No
AttributeValue	String	ByRef	No
Version	String	ByVal	Yes (“~Current~”)
Pp_key1	String	ByVal	Yes (“ ”)
Pp_key2	String	ByVal	Yes (“ ”)
Pp_key3	String	ByVal	Yes (“ ”)
Pp_key4	String	ByVal	Yes (“ ”)
Pp_key5	String	ByVal	Yes (“ ”)

Set1<Object>Attribute

Sets the value of the specified attribute of <Object> with <Object> in the <Object> types given in the list at the beginning of this paragraph.

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
<Object Key>	String	ByVal	No
AttributeName	String	ByVal	No
AttributeValue	String	ByVal	No
Version (for configuration objects only)	String	ByVal	Yes (the default value depends on the Object Key)
Why	String	ByVal	Yes (“ ”)

Set1PpAttribute

Sets the value of the specified attribute of a parameter profile

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
ParameterProfile	String	ByVal	No
AttributeName	String	ByVal	No
AttributeValue	String	ByRef	No
Version	String	ByVal	Yes (“~Highest~”)
Why	String	ByVal	Yes (“ ”)
Pp_key1	String	ByVal	Yes (“ ”)
Pp_key2	String	ByVal	Yes (“ ”)
Pp_key3	String	ByVal	Yes (“ ”)
Pp_key4	String	ByVal	Yes (“ ”)
Pp_key5	String	ByVal	Yes (“ ”)

3.7.5 Get/Set1<xx><yy>Attribute Methods

The following <xx><yy> are supported:

- stip
- stpp
- rtst
- rtp
- pppr
- prmt
- ipie
- rtip

Get1<xx><yy>Attribute

Gets the value of the specified attribute of <xx><yy> with <xx> and <yy> (<xx> must not be 'pp'). The method will look at the first occurrence of <yy> in <xx> and call the API **unapiprp.GetUsedObjectAttribute**.

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
<xx>	String	ByVal	No
<yy>	String	ByVal	No
AttributeName	String	ByVal	No
AttributeValue	String	ByRef	No
Version	String	ByVal	Yes (the default value depends on the Object Key)

Get1<xx>PpAttribute

Gets the value of the specified attribute of a parameter profile in <xx>. The method will call the correct API (e.g. **unapist.GetStPpAttribute**)

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
<xx>	String	ByVal	No
pp	String	ByVal	No
AttributeName	String	ByVal	No
AttributeValue	String	ByRef	No
Version	String	ByVal	Yes ("~Current~")
Pp_key1	String	ByVal	Yes (" ")
Pp_key2	String	ByVal	Yes (" ")
Pp_key3	String	ByVal	Yes (" ")
Pp_key4	String	ByVal	Yes (" ")
Pp_key5	String	ByVal	Yes (" ")

Get1PpPrAttribute

Gets the value of the specified attribute of a parameter profile in <xx>. The method will call the API **unapihpp.GetPpPrAttribute**.

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
ParameterProfile	String	ByVal	No
pr	String	ByVal	No
AttributeName	String	ByVal	No
AttributeValue	String	ByRef	No
Version	String	ByVal	Yes (“~Current~”)
Pp_key1	String	ByVal	Yes (“ “)
Pp_key2	String	ByVal	Yes (“ “)
Pp_key3	String	ByVal	Yes (“ “)
Pp_key4	String	ByVal	Yes (“ “)
Pp_key5	String	ByVal	Yes (“ “)

Set1<xx><yy>Attribute

Sets the value of the specified attribute of <yy> in <xx> (<xx> must not be ‘pp’). The method will call the API **unapiprp.SaveUsedObjectAttribute**.

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
<xx>	String	ByVal	No
<yy>	String	ByVal	No
AttributeName	String	ByVal	No
AttributeValue	String	ByVal	No
Version	String	ByVal	Yes (the default value depends on the Object Key)
Why	String	ByVal	Yes (“”)

Set1<xx>PpAttribute

Sets the value of the specified attribute of a parameter profile

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
<xx>	String	ByVal	No
pp	String	ByVal	No
AttributeName	String	ByVal	No
AttributeValue	String	ByRef	No
Version	String	ByVal	Yes (“~Highest~”)
Why	String	ByVal	Yes (“”)
Pp_key1	String	ByVal	Yes (“ “)
Pp_key2	String	ByVal	Yes (“ “)

Name	Type	ByVal/ByRef	Optional (Default Value)
Pp_key3	String	ByVal	Yes (" ")
Pp_key4	String	ByVal	Yes (" ")
Pp_key5	String	ByVal	Yes (" ")

Set1PpPrAttribute

Sets the value of the specified attribute of a parameter in a parameter profile

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
ParameterProfile	String	ByVal	No
pr	String	ByVal	No
AttributeName	String	ByVal	No
AttributeValue	String	ByRef	No
Version	String	ByVal	Yes ("~Highest ~")
Why	String	ByVal	Yes ("")
Pp_key1	String	ByVal	Yes (" ")
Pp_key2	String	ByVal	Yes (" ")
Pp_key3	String	ByVal	Yes (" ")
Pp_key4	String	ByVal	Yes (" ")
Pp_key5	String	ByVal	Yes (" ")

3.7.6 Get/Set1<Object>GroupKey Methods

The following <xx> types are supported:

- rq
- sc
- me
- st
- rt

Get1<Object>GroupKey

Gets the value of the specified group key of <Object>.

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
<Object Key>	String	ByVal	No
GroupKeyName	String	ByVal	No
GroupKeyValue	String	ByRef	No
Version (only for configuration objects)	String	ByVal	Yes (the default value depends on the Object Key)

Set1<Object>GroupKey

Sets the value of the specified group key of <Object>.

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
<Object Key>	String	ByVal	No
GroupKeyName	String	ByVal	No
GroupKeyValue	String	ByVal	No
Version (only for configuration objects)	String	ByVal	Yes (the default value depends on the object Key)
Why	String	ByVal	Yes ("")

3.7.7 Get/Set1<Object>Result Methods

The following <Object> types are supported:

- pg
- pa
- me
- ce
- rqii
- scii.

Get1<Object = Pg / Me / Ce>Result

Gets the result values of the specified <Object>.

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
Value_f	Double	ByRef	Yes ()
Value_s	String	ByRef	Yes ("")
Unit	String	ByRef	Yes ("")

Get1<Object = Pa>Result

Gets the result values of the specified parameter.

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
SampleCode	String	ByVal	No
ParameterGroup	String	ByRef	Yes ("")
Parameter	String	ByVal	Yes ("")
Value_f	Double	ByRef	Yes (NULL_DOUBLE)
Value_s	String	ByRef	Yes ("")
Unit	String	ByRef	Yes ("")

Name	Type	ByVal/ByRef	Optional (Default Value)
ValidSpecsA	String	ByRef	Yes (")
ValidSpecsB	String	ByRef	Yes (")
ValidSpecsC	String	ByRef	Yes (")
ValidLimitsA	String	ByRef	Yes (")
ValidLimitsB	String	ByRef	Yes (")
ValidLimitsC	String	ByRef	Yes (")
ValidTargetA	String	ByRef	Yes (")
ValidTargetB	String	ByRef	Yes (")
ValidTargetC	String	ByRef	Yes (")

Get1<Object = Rqli / Scli>Result

Gets the result values of the specified info field.

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
InfoCard	String	ByRef	Yes (")
InfoField	String	ByVal	Yes (")
lvalue	String	ByRef	Yes (")

Set1<Object = Pg / Pa / Me / Ce>Result

Sets the result values of the specified <Object>.

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
Value_f	Double	ByVal	Yes (NULL_DOUBLE)
Value_s	String	ByVal	Yes (")
Unit	String	ByVal	Yes (")
Executor	String	ByVal	Yes (")
Why	String	ByVal	Yes (")

Set1<Object = Rqli / Scli>Result

Sets the result values of the specified <Object>.

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
InfoCard	String	ByRef	Yes (")
InfoField	String	ByVal	Yes (")
lvalue	String	ByVal	Yes (")

3.7.8 Get/Set1<Object>Spec Methods

The following <Object> type is supported:

- pa

Get1PaSpec

Gets/Sets one specification value in the specifications set (SpecSet) of the specified parameter.

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
SpecSet	String	ByVal	No
SampleCode	String	ByVal	No
ParameterGroup	String	ByVal	No
Parameter	String	ByVal	No
SpecName	String	ByVal	No
SpecVal	String	ByRef	No

3.7.9 GetAll<Object>Results Methods

The following <Object> types are supported:

- pa
- rqii
- scii.

GetAll<Object = Pa>Results

Gets the result values of the specified parameters.

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
SampleCode	String	ByVal	No
ParameterGroup	String	ByVal	Yes ("")
AllPaResults	Variant	ByRef	Yes ()

Remark: <AllPaResults> is a 2-dimensional array with the following structure:

	Pg	Pa	Value_s	Value_f	Unit	ValidSpecsA	ValidSpecsB	ValidSpecsC	ValidLimitsA	ValidLimitsB	ValidLimitsC	ValidTargetA	ValidTargetB	ValidTargetC
Pa1	0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	0,10	0,11	0,12	0,13

Pa2	1,0	1,1	1,2	1,3	1,4	1,5	1,6	1,7	1,8	1,9	1,10	1,11	1,12	1,13
Pa...	...,0	...,1	...,2	...,3	...,4	...,5	...,6	...,7	...,8	...,9	...,10	...,11	...,12	...,13

GetAll<Object = Rqli / Scli>Results

Gets the result values of the specified infofields.

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
InfoCard	String	ByVal	Yes (")
AllliResults	Variant	ByRef	Yes ()

Remark: <AllliResults> is a 2-dimentional array with the following structure:

	lc	li	lvalue
li1	0,0	0,1	0,2
li2	1,0	1,1	1,2
li...	...,0	...,1	...,2

3.7.10 Get<Object>List Methods

The following <Object> types are supported:

- pa
- rqii
- scii.

Get<Object = Pa>List

Gets a list of sample parameters.

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
SampleCode	String	ByVal	No
ParameterGroup	String	ByVal	Yes (")
PaList	Variant	ByRef	Yes ()

Remark: <PaList> is a 2-dimentional array with the following structure:

	pg	pa
Pa1	0,0	0,1
Pa2	1,0	1,1
Pa...	...,0	...,1

Get<Object = Rqli / Scli>List

Gets a list of request or sample info fields.

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
RequestCode / SampleCode	String	ByVal	No
InfoCard	String	ByVal	Yes ("")
liList	Variant	ByRef	Yes ()

Important

<liList> is a 2-dimentional array with the following structure:

Remark: <liList> is a 2-dimentional array with the following structure:

	lc	li
li1	0,0	0,1
li2	1,0	1,1
li...	...,0	...,1

3.7.11 Get<xx><yy>List Methods

Supported <xx><yy>: stip, stpp, rtst, rtp, pppr, prmt, ipie, mtccl.

Gets a list of assigned <yy> to a <xx>.

The table below lists the parameters for methods where <xx> and <yy> is not 'pp'.

Name	Type	ByVal/ByRef	Optional (Default Value)
<xx>	String	ByVal	No
<yy>List	Variant	ByRef	No
Version (only for configuration objects)	String	ByVal	Yes (the default value depends on the object Key)

<yy>List is a 2-dimentional array with the following structure:

	<yy>	<yy>_version
Obj1	lp1	1.0
Obj2	lp2	1.1
<yy>...	...,0	...,1

Get<xx>PpList methods

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
<xx>	String	ByVal	No
PpList	Variant	ByRef	No
Version	String	ByVal	Yes (the default value depends on the object Key)

<PpList> is a 2-dimentional array with the following structure:

	pp	pp_version	Pp_key1	Pp_key2	Pp_key3	Pp_key4	Pp_key5
Obj1	Pp1	1.0					
Obj2	Pp2	1.1					
<yy>...	..,0	..,1					

GetPpPrList method

Gets a list of assigned parameters to a parameter profile.

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
Pp	String	ByVal	No
prList	Variant	ByRef	No
Version	String	ByVal	Yes ("~Current~")
Pp_key1	String	ByVal	Yes (" ")
Pp_key2	String	ByVal	Yes (" ")
Pp_key3	String	ByVal	Yes (" ")
Pp_key4	String	ByVal	Yes (" ")
Pp_key5	String	ByVal	Yes (" ")

prList is a 2-dimentional array with the following structure:

	pr	pr_version
Obj1	Pr1	1.0
Obj2	pr2	1.1
<yy>...	..,0	..,1

3.7.12 GetRqSample method

Gets a list of samples belonging to a specified request.

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
RequestCode	String	ByVal	No
Sc	Variant	ByRef	No
St	Variant	ByRef	No
St_Version	Variant	ByRef	Yes (" ")
Description	Variant	ByRef	Yes (" ")
Assign_date	Variant	ByRef	Yes (" ")
Assigned_by	Variant	ByRef	Yes (" ")

3.8 Event Methods

The **Unilab** interface can handle the following event methods:

- Enabling/Disabling Events
- EventOccurred Event
- EventOccurredExt Event

3.8.1 Enabling/Disabling Events

EnableEvents

Registers a client for Unilab events. Returns a reference to a unique instance of the EventService class. If the instance does not yet exist, it will be created first.

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
EventServiceName	String	ByRef	Yes
ChangePermanently	Boolean	ByVal	Yes

DisableEvents

Unregisters a client for Unilab events. If all clients for the specified ServiceName are unregistered, the corresponding instance of the EventService object will be destroyed.

Parameters: None

3.8.2 EventOccurred Event

Occurs whenever new event records are available in the UTEVSINK table on the Unilab DB.

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
EventName	String	ByVal	No
EventDetails	String	ByVal	No

3.8.3 EventOccurredExt Event

Occurs at the same time the **EventOccurred** event is fired, but provides extended information about the event.

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
EventName	String	ByVal	No
EventDetails	String	ByVal	No
ObjectType	String	ByVal	No
ObjectId	String	ByVal	No
ObjectStatus	String	ByVal	No
ObjectLifeCycle	String	ByVal	No
ObjectLifeCycleVersion	String	ByVal	No
ClientId	String	ByVal	No
Application	String	ByVal	No
DbApiName	String	ByVal	No
UserName	String	ByVal	No
TransactionSequence	Long	ByVal	No
EventSequence	Long	ByVal	No

Important

The client has no way to control which events are raised. This is defined in the Unilab DB.

3.9 User Management Methods

CreateUser

This function calls the APIs **unapiad.saveAddress** and **unapius.createuser**.

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
a_user	String	ByVal	No
a_identification_type	String	ByVal	No
a_identified_by_string	String	ByVal	No
a_user_prof	Long	ByVal	No
a_person	String	ByVal	Yes ("")

DeleteUser

This function calls the API **unapius.deleteuser**.

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
a_user	String	ByVal	No
a_user_prof	long	ByVal	No

Get1UsProperty

Returns values from **unapiad.getaddress**.

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
User	String	ByVal	No
PropertyName	String	ByVal	No
PropertyValue	Variant	ByRef	No

Set1UsProperty

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
User	String	ByVal	No
PropertyName	String	ByVal	No
PropertyValue	Variant	ByRef	No
Why	String	ByVal	Yes ("")

Get1UpProperty

Returns values from **unapiup.getuserprofile**.

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
UserProfile	String	ByVal	No
PropertyName	String	ByVal	No
PropertyValue	Variant	ByRef	No

Set1UpProperty

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
UserProfile	String	ByVal	No
PropertyName	String	ByVal	No
PropertyValue	Variant	ByRef	No
Why	String	ByVal	Yes ("")

DeleteUserProfile

The table below lists the parameters.

Name	Type	ByVal/ByRef	Optional (Default Value)
a_user_profile	String	ByVal	No
Why	String	ByVal	Yes ("")

4 Object Properties

4.1 Request Properties

The table below give an overview of the request (rq) properties.

<xx = rq> : PropertyNames			
Rq	(read-only)	allow_any_st	
Rt		allow_new_sc	
rt_version		responsible	
Description		sc_counter	(read-only)
descr_doc		last_comment	(read-only)
descr_doc_version		rq_class	
sampling_date		log_hs	
creation_date		log_hs_details	
created_by		Allow_modify	(read-only)
exec_start_date		active	(read-only)
exec_end_date		Lc	
due_date		lc_version	
Priority		Ss	(read-only)
label_format		lc_description	(read-only)
date1		ss_description	(read-only)
date2		GroupKeyValues	(read-only)
date3		Ar	(read-only)
date4		AttributeValues	(read-only)
date5		Changed	(read-only)

4.2 Sample Properties

The table below give an overview of the sample (sc) properties

<xx = sc> : PropertyNames			
Sc	(read-only)	date2	
St		date3	
st_version		date4	
Description		date5	
shelf_life_val		allow_any_pp	
shelf_life_unit		last_comment	(read-only)
sampling_date		sc_class	
creation_date		log_hs	

<xx = sc> : PropertyNames			
created_by		log_hs_details	
exec_start_date		allow_modify	(read-only)
exec_end_date		active	(read-only)
Priority		lc	
label_format		Lc_version	
descr_doc		ss	(read-only)
descr_doc_version		lc_description	(read-only)
Rq		ss_description	(read-only)
date1		sd	
Ar	(read-only)	GroupKeyValues	(read-only)
AttributeValues	(read-only)	Changed	(read-only)

4.3 ParameterGroup Properties

The table below give an overview of the parameter group (pg) properties.

<xx = pg> : PropertyNames			
Sc	(read-only)	assigned_by	
Pg	(read-only)-	manually_added	
PgNode	(read-only)	Format	
Pp_version	(read-only)	confirm_assign	
Pp_key1	(read-only)	Allow_any_pr	
Pp_key2	(read-only)	never_create_methods	
Pp_key3	(read-only)	Delay	
Pp_key4	(read-only)	Delay_unit	
Pp_key5	(read-only)	reanalysis	(read-only)
Description		last_comment	(read-only)
value_f		pg_class	
value_s		log_hs	
Unit		log_hs_details	
exec_start_date		allow_modify	(read-only)
exec_end_date		active	(read-only)
executor		lc	
Planned_executor		lc_version	
manually_entered		ss	(read-only)
assign_date		lc_description	(read-only)
AttributeValues	(read-only)	ss_description	(read-only)
Changed	(read-only)	Id	(read-only)

4.4 Parameter Properties

The table below give an overview of the parameter (pa) properties.

<xx = pa> : PropertyNames			
Sc	(read-only)	calc_method	
Pg	(read-only)	calc_cf	
PgNode	(read-only)	alarm_order	
Pa	(read-only)	valid_specsa	
PaNode	(read-only)	valid_specsb	
Pr_version		valid_specsc	
Description		valid_limitsa	
value_f		valid_limitsb	
value_s		valid_limitsc	
Unit		valid_targeta	
exec_start_date		valid_targetb	
exec_end_date		valid_targetc	
executor		log_exceptions	
Planned_executor		reanalysis	(read-only)
manually_entered		last_comment	(read-only)
assign_date		pa_class	
assigned_by		log_hs	
manually_added		log_hs_details	
Format		allow_modify	(read-only)
td_info		active	(read-only)
td_info_unit		lc	
confirm_uid		lc_version	
allow_any_me		ss	(read-only)
Delay		lc_description	(read-only)
delay_unit		ss_description	(read-only)
min_nr_results		ld	(read-only)
AttributeValues	(read-only)	SpecSet	
Changed	(read-only)	SpecValue	(read-only)

4.5 Method Properties

The table below give an overview of the method (me) properties.

<xx = me> : PropertyNames			
Sc	(read-only)	Format	
Pg	(read-only)	accuracy	
PgNode	(read-only)	real_cost	
Pa	(read-only)	real_time	
PaNode	(read-only)	calibration	
Method_class	(read-only)	Confirm_complete	
menode	(read-only)	autorecalc	
Mt_version		method_result_editable	
description		next_cell	
value_f		sop	
value_s		sop_version	
unit		plaus_low	
exec_start_date		plaus_high	
exec_end_date		winsize_x	
executor		winsize_y	
Eq		reanalysis	(read-only)
Eq_version		last_comment	(read-only)
planned_executor		Method	(read-only)
planned_eq		log_hs	
planned_eq_version		log_hs_details	
manually_entered		allow_modify	
allow_add		active	
Assign_date		lc	
assigned_by		lc_version	
manually_added		ss	
Lab		lc_description	(read-only)
Delay		ss_description	(read-only)
delay_unit		AttributeValues	(read-only)
Changed	(read-only)	GroupKeyValues	(read-only)
Id	(read-only)		

4.6 Cell Properties

The table below give an overview of the cell (ce) properties.

<xx = ce> : PropertyNames			
Sc	(read-only)	Winspace_x	
Pg	(read-only)	Winspace_y	
PgNode	(read-only)	is_protected	
Pa	(read-only)	mandatory	
PaNode	(read-only)	Hidden	
method	(read-only)	Format	
menode	(read-only)	Eq	
reanalysis	(read-only)	eq_version	
cell	(read-only)	component	
cellnode	(read-only)	calc_tp	
dsp_title		calc_formula	
value_f		valid_cf	
Value_s		max_x	
cell_tp		max_y	
pos_x		multi_select	
pos_y		unit	
Align		Changed	(read-only)
Id	(read-only)		

4.7 Request InfoCard Properties

The table below give an overview of the request info card (rqic) properties.

<xx = rqic> : PropertyNames			
Rq	(read-only)	last_comment	(read-only)
Ic	(read-only)	ic_class	
icnode	(read-only)	log_hs	
lp_version		log_hs_details	
description		allow_modify	(read-only)
winspace_x		active	(read-only)
winspace_y		Lc	
is_protected		lc_version	
Hidden		ss	(read-only)
manually_added		lc_description	(read-only)
next_ii		ss_description	(read-only)
AttributeValues	(read-only)	Id	(read-only)
Changed	(read-only)	lp_version	
Sc			

4.8 Request InfoField Properties

The table below give an overview of the request info field (rqii) properties.

<xx = rqii> : PropertyNames			
Rq	(read-only)	dsp_len	
lc	(read-only)	dsp_tp	
icnode	(read-only)	dsp_rows	
li	(read-only)	ii_class	
liNode	(read-only)	log_hs	
le_version		log_hs_details	
iivalue		allow_modify	(read-only)
pos_x		active	(read-only)
pos_y		Lc	
is_protected		lc_version	
Mandatory		ss	(read-only)
Hidden		lc_description	(read-only)
dsp_title		ss_description	(read-only)
Changed	(read-only)	Id	(read-only)

4.9 Sample InfoCard Properties

The table below give an overview of the sample info card (scic) properties

<xx = scic> : PropertyNames			
Sc	(read-only)	last_comment	(read-only)
lc	(read-only)	ic_class	
icnode	(read-only)	log_hs	
lp_version		log_hs_details	
description		allow_modify	(read-only)
winsize_x		active	(read-only)
winsize_y		Lc	
is_protected		lc_version	
Hidden		ss	(read-only)
manually_added		lc_description	(read-only)
next_ii		ss_description	(read-only)
AttributeValues	(read-only)	Id	(read-only)
Changed	(read-only)	lp_version	

4.10 Sample InfoField Properties

The table below give an overview of the sample infofield (scii) properties.

<xx = scii> : PropertyNames			
Sc	(read-only)	dsp_len	

<xx = scii> : PropertyNames			
lc	(read-only)	dsp_tp	
icnode	(read-only)	dsp_rows	
li	(read-only)	ii_class	
liNode	(read-only)	log_hs	
le_version		log_hs_details	
livalue		allow_modify	(read-only)
pos_x		active	(read-only)
pos_y		Lc	
is_protected		lc_version	
Mandatory		ss	(read-only)
Hidden		lc_description	(read-only)
dsp_title		ss_description	(read-only)
Changed	(read-only)	Id	(read-only)

5 Return Codes and Error Numbers

5.1 Return Codes

The table below lists the return codes.

Return Code ID	Description	Return Code
OLRET_SUCCESS	Execution of method was successful	0
OLRET_ERROROCCURRED	Error raised during execution of method	-1
OLRET_TRANSACTIONBUSY	There is already a transaction busy, no new transaction was started	-2
OLRET_NOTTRANSACTIONBUSY	There is no transaction busy	-3

5.2 Error Numbers

Unilab object errors

The table below lists the Unilab object errors.

Unilab Object Error ID	Error Description	Error Number
OLERR_INVALIDPARAMETERVALUE	Invalid parameter value	-2147216503
OLERR_READONLYPROPERTY	Property is read-only	-2147216502
OLERR_NODATAFOUND	No data found	-2147216501
OLERR_OBJECTDOESNOTEXIST	Object does not exist	-2147216500

Database errors

The table below lists the database errors.

DB Error ID		Error Description	Error Number
DBERR_GENFAIL	1	General Failure	-2147211503
DBERR_NOOBJECT	2	Object does not exist	-2147211502
DBERR_NOACCESS	3	No access to object	-2147211501
DBERR_READONLY	4	Read-only access to object	-2147211500
DBERR_NOTMODIFIABLE	5	Object is not modifiable	-2147211499
DBERR_NOTACTIVE	6	Object is not active	-2147211498
DBERR_INUSE	7	Object is used by other objects (referential integrity violation)	-2147211497
DBERR_INTXN	8	Transaction in progress	-2147211496
DBERR_NOOBJTYP	9	Object type not specified	-2147211495
DBERR_NOOBJID	10	Object ID not specified	-2147211494

DB Error ID		Error Description	Error Number
DBERR_NORECORDS	11	No records found	-2147211493
DBERR_NROFROWS	12	Invalid number of rows specified	-2147211492
DBERR_NOCURSOR	13	Cursor not open	-2147211491
DBERR_TRANSITION	14	Object is in transition	-2147211490
DBERR_WHERECLAUSE	15	Invalid 'where' clause specified	-2147211489
DBERR_OBJECTLCMATCH	16	Object life cycle does not match	-2147211488
DBERR_OBJECTSSMATCH	17	Object status does not match	-2147211487
DBERR_NOTAUTHORISED	18	Not authorized to make any modifications	-2147211486
DBERR_NOLC	19	Life cycle model does not exist	-2147211485
DBERR_BTXTN	20	BeginTxn called twice	-2147211484
DBERR_TXNNOTBEGUN	21	EndTxn called before BeginTxn	-2147211483
DBERR_NOUSEDOBJECT	22	Used object does not exist	-2147211482
DBERR_NOTFOUND	23	Object related data could not be found	-2147211481
DBERR_UP	24	Invalid user profile	-2147211480
DBERR_SYSDEFAULTS	25	System defaults not set	-2147211479
DBERR_NEXTROWS	26	Invalid value in next rows argument	-2147211478
DBERR_OBJTP	27	Missing or invalid user profile ID	-2147211477
DBERR_PARTIALSAVE	28	Not all the specified items in the array are saved	-2147211476
DBERR_ALLOWANYPP	29	Invalid value in 'allow_any_pp' column	-2147211475
DBERR_MOD_UPDATE	30	Partial update: this record could not be updated	-2147211474
DBERR_MOD_INSERT	31	Partial update: this record could not be inserted	-2147211473
DBERR_MOD_DELETE	32	Partial update: this record could not be deleted	-2147211472
DBERR_NOMAINOBJ	33	Main object for method cell output cannot be found	-2147211471
DBERR_FLOATTOSTRING	34	Float value could not be converted to a string	-2147211470
DBERR_INVALIDMODFLAG	35	Invalid value in 'modify_flag' column	-2147211469
DBERR_NOTIMPLEMENTED	36	Specified cell output type not (yet) supported	-2147211468
DBERR_ALLOWANYPR	37	Invalid value in 'allow_any_pr' column	-2147211467
DBERR_OUTOFCALENDAR	38	The table 'utweeknr' must be extended to the future	-2147211466
DBERR_NOSS	39	The specified status definition does not exist	-2147211465
DBERR_RESERVEDID	40	This id is reserved for system purposes	-2147211464
DBERR_EVMGRSTARTNOT AUTHORISED	41	User is not authorized to start/stop the event manager	-2147211463
DBERR_EVMGRNOTSTARTED	42	Event manager not running	-2147211462
DBERR_INVALIDDATE	43	Invalid date	-2147211461
DBERR_DATEMULTIPLMATCH	44	The date format mask is not detailed enough	-2147211460
DBERR_INVALIDDATEFORMAT	45	The date format is not valid	-2147211459
DBERR_OPACTIVE	50	The object is active and cannot be deleted	-2147211454

DB Error ID		Error Description	Error Number
DBERR_NEWNODENOTZERO	51	Attempt to insert new object with non-zero node	-2147211453
DBERR_NODELIMITOVERF	52	Node overflow (impossible to insert node)	-2147211452
DBERR_ALLOWANYMT	53	Invalid value in 'allow_any_mt' column	-2147211451
DBERR_NOTINTRANSACTION	54	The API function has not been called inside a transaction	-2147211450
DBERR_DD	55	Invalid data domain, must be in ('1', '2', ..., '16')	-2147211449
DBERR_UPONE	56	User profile 1 cannot be deleted	-2147211448
DBERR_UPINUSE	57	This user profile is in use as default user profile	-2147211447
DBERR_INVALIDKEY	58	Invalid license key	-2147211446
DBERR_TOOMANYUSERS	59	More users connected than allowed by your license key	-2147211445
DBERR_LICEXPIRED	60	License key expired	-2147211444
DBERR_RECURSIVEDATA	61	recursivity in configuration not allowed	-2147211443
DBERR_ALREADYEXISTS	62	Target clone already exists	-2147211442
DBERR_TRSEQ	63	Invalid value in transaction sequence argument	-2147211441
DBERR_OBJECTLCVERSIONMATCH	64	Object life cycle version does not match	-2147211440
DBERR_VALUelistTP	70	Invalid value list type	-2147211434
DBERR_DEFVALUETP	71	Invalid default value type	-2147211433
DBERR_AULEVEL	72	Invalid attribute level	-2147211432
DBERR_INHERITAU	73	Invalid value in 'inherit attribute' flag	-2147211431
DBERR_LOGHS	74	Invalid value in 'audit trail logging' flag	-2147211430
DBERR_TEMPLATE	75	Invalid value in 'template' flag	-2147211429
DBERR_PROTECTED	76	Invalid value in 'protected' flag	-2147211428
DBERR_HIDDEN	77	Invalid value in 'hidden' flag	-2147211427
DBERR_ALLOWMODIFY	78	Invalid value in 'allow modify' flag	-2147211426
DBERR_ACTIVE	79	Invalid value in 'active' flag	-2147211425
DBERR_AR	80	Invalid value in access rights	-2147211424
DBERR_INHERITGK	81	Invalid value in 'inherit_gk' column	-2147211423
DBERR_NOCHANGE	82	No change was done	-2147211422
DBERR_NODEFVALUE	83	No default value found	-2147211421
DBERR_UPPREF	84	Invalid user profile preference	-2147211420
DBERR_SELCOLSINVALID	85	Invalid or empty value in 'col_id' column	-2147211419
DBERR_NOWRITEACCESS	86	No write access to object (modifications not allowed)	-2147211418
DBERR_USERNOTACTIVE	87	User is not active	-2147211417
DBERR_UPNOTACTIVE	88	User profile is not active	-2147211416
DBERR_SELECTED	89	Invalid value in 'selected' column	-2147211415
DBERR_MISSINGPRMT	90	Missing parameter-method link	-2147211414
DBERR_MULTISELECT	91	Invalid value in 'multi_select' column	-2147211413
DBERR_INVALIDNAME	92	User name should only contain	-2147211412

DB Error ID		Error Description	Error Number
		alphanumeric characters and underscore(s)	
DBERR_AUTOREFRESH	93	Invalid autorefresh	-2147211411
DBERR_LOGHSDetails	94	Invalid value in 'audit trail details logging'	-2147211410
DBERR_UQSHORTCUTKEY	100	Shortcut key is not unique	-2147211404
DBERR_NOAUNAME	110	Attribute name must be specified	-2147211394
DBERR_SINGLEVALUED	111	Invalid value in 'single valued' column	-2147211393
DBERR_NEWVALALLOWED	112	Invalid value in 'new values allowed' column	-2147211392
DBERR_STOREDB	113	Invalid value in 'store in database' column	-2147211391
DBERR_RUNMODE	114	Invalid value in run mode	-2147211390
DBERR_UC	120	Invalid unique code mask	-2147211384
DBERR_EDITALLOWED	121	Invalid value in 'edit allowed' column	-2147211383
DBERR_FIXEDLENGTH	122	Invalid value in 'fixed_length' column	-2147211382
DBERR_UCSTRUCTURE	123	Semantic error in code mask structure	-2147211381
DBERR_SEQMAXMINVALUE	124	'high_cnt' must be greater than 'low_cnt'	-2147211380
DBERR_SEQINCR	125	Increment is too large (for specified range)	-2147211379
DBERR_HIGHCNTNOTNULL	126	The high count value can not be null	-2147211378
DBERR_UQDEFMASKFOR	127	There is already a default mask for samples or requests	-2147211377
DBERR_DSPTP	130	Invalid display type	-2147211374
DBERR_LU	131	Invalid value in lookup pointer	-2147211373
DBERR_DATATP	132	Invalid value in data type	-2147211372
DBERR_MANDATORY	133	Invalid value in mandatory flag	-2147211371
DBERR_UQLU	134	Uniqueness of lookup entries not respected	-2147211370
DBERR_UQSS	135	Uniqueness of states not respected	-2147211369
DBERR_TDDELAY	170	Invalid value in trending unit	-2147211334
DBERR_CONFIRMUSERID	171	Invalid value in 'confirm user ID' flag	-2147211333
DBERR_CALCMETHOD	172	Invalid calculation of parameter result	-2147211332
DBERR_DELAYUNIT	173	Invalid value in delay unit	-2147211331
DBERR_LOGEXCEPTIONS	174	Invalid value in 'log exceptions' flag	-2147211330
DBERR_ALLOWADD	175	Invalid value in 'allow additional measurements' flag	-2147211329
DBERR_IGNOREOTHER	176	Invalid value in 'ignore other' flag	-2147211328
DBERR_FREQTP	177	Invalid frequency type specified	-2147211327
DBERR_INVERTFREQ	178	Invalid value in invert frequency	-2147211326
DBERR_STFREQ	179	Invalid value in 'sample type based freq' flag	-2147211325
DBERR_CONFIRMASSIGN	190	Invalid value in 'confirm assignment' flag	-2147211314
DBERR_ADDMISSINGPR	191	Invalid value in 'add missing parameters' flag	-2147211313
DBERR_REMOVEEXTRACPR	192	Invalid value in 'remove extra parameters' flag	-2147211312
DBERR_SPECSET	193	The specified specification set does not exist	-2147211311
DBERR_FREQVAL	194	Invalid or missing frequency value	-2147211310
DBERR_LASTCNT	195	Invalid or missing last count value	-2147211309
DBERR_FREQUNIT	196	Invalid frequency unit	-2147211308

DB Error ID		Error Description	Error Number
DBERR_ISPP	197	Invalid value in 'ispp' flag	-2147211307
DBERR_NEVERCREATEMETHODS	198	Invalid value in 'never create methods' flag	-2147211306
DBERR_AUTORECALC	210	Invalid value for 'autorecalc' flag	-2147211294
DBERR_CALIBRATION	211	Invalid value in calibration flag	-2147211293
DBERR_CELLTP	212	Invalid value for method cell type	-2147211292
DBERR_INPUTTP	213	Invalid value in input type	-2147211291
DBERR_ALIGN	214	Invalid value in alignment	-2147211290
DBERR_CREATENEW	215	Invalid value in 'create new' flag	-2147211289
DBERR_SAVETP	216	Invalid value in save type	-2147211288
DBERR_CIRCULAR	217	Invalid value for 'circular' flag	-2147211287
DBERR_CONFIRMCOMPLETE	218	Invalid value for 'confirm completion' flag	-2147211286
DBERR_NOTCURRENTMETHOD	219	Required action can not happen on historical method results. A reanalysis took place	-2147211285
DBERR_AUTOCREATECELLS	220	Invalid value in 'create cells automatically'	-2147211284
DBERR_MERESULTEDITABLE	221	Invalid value in 'result editable' flag	-2147211283
DBERR_CHGPWD	230	Invalid value in 'change password' flag	-2147211274
DBERR_DEFINEMENU	231	Invalid value in 'define menu' flag	-2147211273
DBERR_CONFIRMCHGSS	232	Invalid value in 'confirm change status' flag	-2147211272
DBERR_SKIPMANDATORY	233	Invalid value in 'allow mandatory fields to be skipped' flag	-2147211271
DBERR_US	234	Invalid user specified	-2147211270
DBERR_FA	235	Invalid functional right specified	-2147211269
DBERR_RULETP	236	Invalid value in rule type	-2147211268
DBERR_RULEVAL	237	Invalid value in rule	-2147211267
DBERR_RULEOK	238	Invalid access right in case rule is OK	-2147211266
DBERR_RULENOK	239	Invalid access right in case rule is not OK	-2147211265
DBERR_TK	240	Invalid or missing task	-2147211264
DBERR_INVALIDUSER	241	The login user ID and the SetConnection value do not match	-2147211263
DBERR_INVALIDUP	242	The User Id in SetConnection does not yield in a valid user profile	-2147211262
DBERR_INHERITFLAG	243	Invalid value in 'inherit' column	-2147211261
DBERR_ISENABLED	244	Invalid value in 'is_enabled' column	-2147211260
DBERR_DBAUSER	245	Action is only allowed to the Unilab DBA user	-2147211259
DBERR_INVALIDDDD	246	Data domain should be between 1 and 16	-2147211258
DBERR_UNIQUEGK	250	Unique constraint of group key was violated	-2147211254
DBERR_QCHECKAU	251	Invalid attribute selected	-2147211253
DBERR_GKSTRUCTEXIST	252	The group key structure already exists	-2147211252
DBERR_PASSWORD	253	Invalid password	-2147211251
DBERR_ENTRYACTION	270	Invalid value in 'entry_action' column	-2147211234
DBERR_ENTRYTP	271	Invalid value in 'entry_tp' column	-2147211233
DBERR_EVENTTP	290	Invalid event type	-2147211214

DB Error ID		Error Description	Error Number
DBERR_EVMGRTP	291	Invalid type of event manager	-2147211213
DBERR_EVMGRPUBIC	292	Invalid value for public event manager	-2147211212
DBERR_USEREXIST	310	The specified user already exists	-2147211194
DBERR_ISUSER	311	Invalid value for 'is_user' flag	-2147211193
DBERR_NOVIEWS	312	No views defined	-2147211192
DBERR_NOPACKAGES	313	No packages defined	-2147211191
DBERR_STRUCTCREATED	315	The group key structure is already created	-2147211189
DBERR_NOTADBAUSER	316	Only DBA users are authorized to perform the required operation (start event manager, equipment manager, ...)	-2147211188
DBERR_COLTP	321	Invalid column type	-2147211183
DBERR_COLASC	322	Invalid value for 'ascending/descending' flag	-2147211182
DBERR_INVALIDWSMODFLAG	350	Invalid value for a_ws_modify_flag argument	-2147211154
DBERR_MAXROWSOVERFLOW	351	The maximum number of rows has been reached in the worksheet	-2147211153
DBERR_SC_COUNTER	352	Invalid value for sc_counter standard property of the worksheet	-2147211152
DBERR_MIN_ROWS	353	Invalid value for min_rows standard property of the worksheet	-2147211151
DBERR_MAX_ROWS	354	Invalid value for max_rows standard property of the worksheet	-2147211150
DBERR_COMPLETE	355	Invalid value for 'complete' flag of the worksheet	-2147211149
DBERR_WSLY	356	The worksheet layout must be specified when defining a worksheet type	-2147211148
DBERR_INVALIDROWNR	357	Invalid row number in the worksheet	-2147211147
DBERR_SC_CREATE	358	Invalid value for 'sc_create' flag of the worksheet	-2147211146
DBERR_WSALREADYEXIST	359	The worksheet already exists	-2147211145
DBERR_NODFLTMASKFORWS	360	There is no default unique code mask defined for worksheets	-2147211144
DBERR_MULTDEFMASKFORWS	361	There are several unique code mask for worksheets	-2147211143
DBERR_ICALREADYEXIST	400	The infocard already exists	-2147211104
DBERR_IIALREADYEXIST	401	The info field already exists	-2147211103
DBERR_MANUALLYADDED	440	Invalid value for 'manually added' flag	-2147211064
DBERR_MANUALLY_ENTERED	441	Invalid value for 'manually entered' flag	-2147211063
DBERR_ADDMETHODSNOTALLOWED	450	'Never create methods' flag is set so no methods can be assigned to the parametergroup	-2147211054
DBERR_NODFLTMASKFORSC	500	There is no default unique code mask for samples	-2147211004
DBERR_MULTDEFMASKFORSC	501	There are several unique code mask for samples	-2147211003
DBERR_SCALREADYEXIST	502	Sample code already exists	-2147211002
DBERR_SELECTEQ	600	Attribute not found for selected equipment	-2147210904

DB Error ID		Error Description	Error Number
DBERR_DETAILSEXIST	601	The method sheet details already exist	-2147210903
DBERR_PLAUSIBILITY	602	Method result is not within plausibility range	-2147210902
DBERR_COMPLETED	603	Invalid value for 'completed' flag	-2147210901
DBERR_MEALREADYEXIST	604	The method already exists	-2147210900
DBERR_RELOWDEV	630	Invalid value for 'relative low deviation' flag	-2147210874
DBERR_RELHIGHDEV	631	Invalid value for 'relative high deviation' flag	-2147210873
DBERR_NORESULT	632	Minimum required number of results is not available	-2147210872
DBERR_ALLOWANYME	633	Invalid value for 'allow any method' flag	-2147210871
DBERR_PGCONFIRMASSIGN	634	Deletion is impossible when the confirmation is already done	-2147210870
DBERR_PGALREADYEXIST	635	The parameter group already exists	-2147210869
DBERR_PAALREADYEXIST	670	The parameter already exists	-2147210834
DBERR_ALLOWANYST	700	Invalid value in 'allow_any_st' column	-2147210804
DBERR_ALLOWNEWSC	701	Invalid value in 'allow_new_sc' column	-2147210803
DBERR_ADDSTPP	702	Invalid value in 'add_stpp' column	-2147210802
DBERR_RQALREADYEXIST	730	The request already exists	-2147210774
DBERR_NODFLTMASKFORRQ	731	No default mask for request	-2147210773
DBERR_MULTDEFMASKFORRQ	732	Multiple default masks for request	-2147210772
DBERR_COPYFLAG	750	Invalid value for a_copy flag argument	-2147210754
DBERR_DELETEFLAG	751	Invalid value for a_delete flag argument	-2147210753
DBERR_NOEOF	752	No end-of-file found in archive file	-2147210752
DBERR_INVALIDSECTION	770	Invalid section	-2147210734
DBERR_NOPARENTOBJECT	771	No parent found for specified object	-2147210733
DBERR_INVALIDROW	772	A row could not be interpreted correctly	-2147210732
DBERR_INVALIDVARIABLE	773	Invalid variable	-2147210731
DBERR_CREATESETTING	774	Invalid value for create flag	-2147210730
DBERR_INVALIDID	775	Invalid object id	-2147210729
DBERR_CAWARNLEVEL	802	Invalid value for calibration warning level	-2147210702
DBERR_CAINPROGRESS	804	Intervention in progress	-2147210700
DBERR_COPYIC	851	Invalid value for a_copy_ic argument	-2147210653
DBERR_COPYPG	852	Invalid value for a_copy_ic argument	-2147210652
DBERR_STOPLCEVALUATION	890	Can be returned by a UNCONDITION function to stop the current life cycle evaluation	-2147210614
DBERR_STOPEVMGR	891	Can be returned by a UNCONDITION/UNACTION function to stop all the event managers in case of problem inside an action	-2147210613
DBERR_NOOBJVERSION	901	Object version not specified	-2147210603
DBERR_NOCURRENTADVERSION	902	No current version found for the specified address	-2147210602
DBERR_NOCURRENTAUVERSION	903	No current version found for the specified attribute	-2147210601

DB Error ID		Error Description	Error Number
DBERR_NOCURRENTEQVERSION	904	No current version found for the specified equipment	-2147210600
DBERR_NOCURRENTIEVERSION	905	No current version found for the specified infofield	-2147210599
DBERR_NOCURRENTIPVERSION	906	No current version found for the specified infoprofile	-2147210598
DBERR_NOCURRENTLCVERSION	907	No current version found for the specified lifecycle	-2147210597
DBERR_NOCURRENTMTVERSION	908	No current version found for the specified method	-2147210596
DBERR_NOCURRENTPPVERSION	909	No current version found for the specified parameterprofile	-2147210595
DBERR_NOCURRENTPRVERSION	910	No current version found for the specified parameter	-2147210594
DBERR_NOCURRENTRTVERSION	911	No current version found for the specified requesttype	-2147210593
DBERR_NOCURRENTSTVERSION	912	No current version found for the specified sampletype	-2147210592
DBERR_NOCURRENTTKVERSION	913	No current version found for the specified task	-2147210591
DBERR_NOCURRENTUPVERSION	914	No current version found for the specified userprofile	-2147210590
DBERR_NOCURRENTWTVERSION	915	No current version found for the specified worksheettype	-2147210589
DBERR_NOCURRENTPTVERSION	916	No current version found for the specified protocol	-2147210588
DBERR_NOCURRENTCYVERSION	917	No current version found for the specified chart type	-2147210587
DBERR_ADVERSION	925	Address version not specified	-2147210579
DBERR_AUVERSION	926	Attribute version not specified	-2147210578
DBERR_EQVERSION	927	Equipment version not specified	-2147210577
DBERR_IEVERSION	928	Infofield version not specified	-2147210576
DBERR_IPVERSION	929	Infoprofile version not specified	-2147210575
DBERR_LCVERSION	930	Lifecycle version not specified	-2147210574
DBERR_MTVERSION	931	Method version not specified	-2147210573
DBERR_PPVERSION	932	Parameterprofile version not specified	-2147210572
DBERR_PRVERSION	933	Parameter version not specified	-2147210571
DBERR_RTVERSION	934	Requesttype version not specified	-2147210570
DBERR_STVERSION	935	Sampletype version not specified	-2147210569
DBERR_TKVERSION	936	Task version not specified	-2147210568
DBERR_UPVERSION	937	Userprofile version not specified	-2147210567
DBERR_WTVERSION	938	Worksheettype version not specified	-2147210566
DBERR_NOTALLOWEDIN21CFR11	939	Not allowed to delete an object because of 21CFR11 standard compliancy	-2147210565
DBERR_PTVERSION	940	Protocol version not specified	-2147210564
DBERR_CYVERSION	941	Chart type version not specified	-2147210563

DB Error ID		Error Description	Error Number
DBERR_PARTIALCHARTSAVE	961	The parameter results are saved but not all datapoints are saved in their charts	-2147210543
DBERR_CHARTNOTMODIFIABLE	962	The chart is not modifiable	-2147210542
DBERR_NULLTIMEPOINT	1001	Timepoint cannot be null	-2147210503
DBERR_TPUNIT	1002	Invalid timepoint unit	-2147210502
DBERR_SDALREADYEXIST	1003	The study code already exists	-2147210501
DBERR_NODFLTMASKFORSD	1004	There is no default code masks for studies	-2147210500
DBERR_MULTDEFMASKFORSD	1005	There is more than one mask specified as default mask for studies	-2147210499