

imc FAMOS Datenbank Kit

Bedienerhandbuch

Doc. Rev.: 1.1 - 09.05.2025



Haftungsausschluss

Diese Dokumentation wurde mit großer Sorgfalt erstellt und auf Übereinstimmung mit der beschriebenen Hard- und Software geprüft. Dennoch können Abweichungen und Fehler nicht ausgeschlossen werden, sodass wir für die vollständige Übereinstimmung keine Gewähr übernehmen.

Technische Änderungen bleiben vorbehalten.

Copyright

© 2025 imc Test & Measurement GmbH, Deutschland

Diese Dokumentation ist geistiges Eigentum von imc Test & Measurement GmbH. imc Test & Measurement GmbH behält sich alle Rechte auf diese Dokumentation vor. Es gelten die Bestimmungen des "imc Software-Lizenzvertrags".

Die in diesem Dokument beschriebene Software darf ausschließlich gemäß der Bestimmungen des "imc Software-Lizenzvertrags" verwendet werden.

Open Source Software Lizenzen

Einige Komponenten von imc-Produkten verwenden Software, die unter der GNU General Public License (GPL) lizenziert sind. Details finden Sie im About-Dialog.

Falls Sie eine Kopie der verwendeten GPL Quellen erhalten möchten, setzen Sie sich bitte mit unserem technischen Support in Verbindung.

Hinweise zu diesem Dokument

Dieses Dokument ist ein Auszug aus der Online-Hilfe von imc FAMOS. Das Handbuch von imc FAMOS ist als chm und als PDF vorhanden. Klicken Sie auf das Symbol , um die Hilfe zu öffnen.

Über das Programm "*imc Hilfe und Dokumentation*" (im Startmenü oder über das Menüband "*Hilfe*" > "*Weitere Dokumente*") erhalten Sie Zugriff auf das PDF-Format und andere Dokumente.

Wie sind die imc FAMOS Dokumente zu lesen?

Wenn Sie bereits mit älteren Versionen von imc FAMOS gearbeitet haben, empfiehlt sich ein Blick in das Thema "*Update-Infos*" (Was ist neu...). Sie erhalten hier einen schnellen Überblick über Neuheiten und Änderungen in der vorliegenden Version.

Erste Schritte

Lesen Sie bitte vor der Installation der Software das Dokument "*Erste Schritte*". Es enthält wesentliche Hinweise zur problemlosen Installation, sowie zur Bedienung.

Alle Informationen aus dem Dokument "*Erste Schritte*" sind auch in dem imc FAMOS Handbuch zu finden.

Handbuch - Komplette Dokumentation

Das Handbuch dient als Nachschlagewerk. Es ist empfehlenswert, später die gesamte Anleitung zu beachten, um viele interessante, aber nicht so offensichtliche Funktionen benutzen zu können.

LiesMich

Sie finden hier Informationen über Änderungen und Erweiterungen in imc FAMOS, die sich nach Drucklegung dieses Handbuches ergeben haben.

Tutorials

Nach erfolgreicher Installation empfehlen wir Ihnen, die Tutorials durchzuarbeiten, um einen ersten Eindruck von imc FAMOS zu erhalten. Es macht Sie in kompakter Form mit der imc FAMOS-Bedienphilosophie und grundlegenden Arbeitsschritten vertraut.

Besondere Hinweise



Warnung

Warnungen enthalten Informationen, die beachtet werden müssen, um den Benutzer vor Schaden zu bewahren bzw. um Sachschäden zu verhindern.



Hinweis

Hinweise bezeichnen nützliche Zusatzinformationen zu einem bestimmten Thema.



Verweis

Verweise sind Hinweise im Text auf eine andere Textstelle.

Inhaltsverzeichnis

- 1 Allgemeine Einführung 5**
 - 1.1 Bevor Sie starten 5
 - 1.2 Technischer Support 5
 - 1.3 Service und Wartung 6
 - 1.4 Rechtliche Hinweise 6
 - 1.5 imc Software-Lizenzvertrag 7
- 2 imc FAMOS Datenbank Kit 10**
 - 2.1 Übersicht 10
 - 2.2 Registrierung in imc FAMOS 10
 - 2.3 Allgemeines 10
 - 2.4 Funktionen 12
 - 2.5 Konvertierung der Daten 34
 - 2.6 Behandlung von Nullwerten 36
 - 2.7 Fehlercodes 37
- Index 42**

1 Allgemeine Einführung

1.1 Bevor Sie starten

Sehr geehrter Nutzer.

1. Die überlassene Software sowie das dazugehörige Handbuch sind für fachkundige und eingewiesene Benutzer ausgestaltet. Sollten sich Unstimmigkeiten ergeben, wenden Sie sich bitte an unseren [technischen Support](#) ⁵.
2. Durch Updates in der fortschreitenden Softwareentwicklung können einzelne Passagen des Handbuchs überholt sein. Wenn Ihnen Abweichungen auffallen, wenden Sie sich bitte an unseren technischen Support.
3. Wenden Sie sich bitte an unseren technischen Support, wenn Sie aufgrund missverständlicher Regelungen oder Ausführungen des vorliegenden Handbuchs zu der Auffassung gelangen, dass Personenschäden zu befürchten sind.
4. Lesen Sie den hier enthaltenen [Lizenzvertrag](#) ⁷. Mit der Nutzung der Software, erkennen Sie die Bedingungen des Lizenzvertrags an.



Hinweis

Hinweis zu den Beschreibungen und Screenshots

- In der Hilfe können auch Anteile enthalten sein, die **gemeinsame imc Softwarekomponenten** beschreiben. Diese Anteile können in Stil und Aufbau von der übrigen Hilfe abweichen. Alle Hilfen sind mit einer Volltextsuche ausgestattet und haben ein Stichwortverzeichnis.
- Die Screenshots in der Dokumentation wurden mit **verschiedenen Windows Versionen** erstellt. Sie können daher vom Erscheinungsbild Ihrer Installation abweichen.

1.2 Technischer Support

Zur technischen Unterstützung steht Ihnen unser technischer Support zur Verfügung:

Telefon: **+49 30 467090-26**

E-Mail: hotline@imc-tm.de

Internet: <https://www.imc-tm.de/service-training/>

Tipps für eine schnelle Bearbeitung Ihrer Fragen:

Sie helfen uns bei Anfragen, wenn Sie die **Seriennummer Ihrer Produkte**, sowie die **Versionsbezeichnung der Software** nennen können. Diese Dokumentation sollten Sie ebenfalls zur Hand haben.

- Die Seriennummer des Gerätes finden Sie z.B. auf dem Typ-Schild auf dem Gerät.
- Die Versionsbezeichnung der Software finden Sie in dem Info-Dialog.

Produktverbesserung und Änderungswünsche

Helfen Sie uns die Dokumentation und die Produkte zu verbessern:

- Sie haben einen Fehler in der Software gefunden oder einen Vorschlag für eine Änderung?
- Das Arbeiten mit dem Gerät könnte durch eine Änderung der Mechanik verbessert werden?
- Im Handbuch oder in den technischen Daten gibt es Begriffe oder Beschreibungen, die unverständlich sind?
- Welche Ergänzungen und Erweiterungen schlagen Sie vor?

Über eine Nachricht an unseren [technischen Support](#) würden wir uns freuen.

1.3 Service und Wartung

Für Service- und Wartungsanfragen steht Ihnen unser Serviceteam zur Verfügung:

Telefon: **+49 30 629396-333** (Mo.-Fr.: 9:00 - 12:00 und 13:00 - 17:00 Uhr)

E-Mail: imc-service@axiometrixsolutions.com

Internet: <https://www.imc-tm.de/service>

Service- und Wartungsarbeiten beinhalten u.a. Kalibrierung und Justage, Service Check, Reparaturen.

1.4 Rechtliche Hinweise

Qualitätsmanagement



imc Test & Measurement GmbH ist seit Mai 1995 DIN EN ISO 9001 zertifiziert und seit November 2023 auch DIN EN ISO 14001. Aktuelle Zertifikate, Konformitätserklärungen und Informationen zu unserem Qualitätsmanagementsystem finden Sie unter: <https://www.imc-tm.de/qualitaetssicherung/> und im [Download Bereich von imc FAMOS](#).

imc Gewährleistung

Es gelten die Allgemeinen Geschäftsbedingungen der imc Test & Measurement GmbH.

Haftungsbeschränkung

Alle Angaben und Hinweise in diesem Dokument wurden unter Berücksichtigung der geltenden Normen und Vorschriften, dem Stand der Technik sowie unserer langjährigen Erkenntnisse und Erfahrungen zusammengestellt. Die Dokumentation wurde auf Übereinstimmung mit der beschriebenen Hard- und Software geprüft. Dennoch können Abweichungen und Fehler nicht ausgeschlossen werden, sodass wir für die vollständige Übereinstimmung keine Gewähr übernehmen. Technische Änderungen bleiben vorbehalten.

Der Hersteller übernimmt keine Haftung für Schäden aufgrund:

- Nichtbeachtung des Handbuchs sowie der Ersten Schritte
- Nichtbestimmungsgemäßer Verwendung.

1.5 imc Software-Lizenzvertrag

imc Test & Measurement GmbH
Voltastraße 5
13355 Berlin
Handelsregister: Berlin-Charlottenburg HRB 28778
Geschäftsführer: Michael John Flaherty, Frank Mayer

imc Test & Measurement GmbH
Bestimmungen
über die Nutzung von Software der imc Test & Measurement GmbH
Stand: 18.01.2024

§ 1 Vertragsgegenstand

- (1) Diese Bestimmungen gelten ergänzend zu den "Allgemeinen Geschäftsbedingungen über Lieferungen und Leistungen der imc Test & Measurement GmbH an Kunden" für alle Verträge mit der imc Test & Measurement GmbH ("imc"), die die Überlassung von Nutzungsrechten an jedweder von imc erstellter Software (Standard-Software, kundenspezifisch erstellte oder angepasste Software, die auf den maschinenlesbaren Trägern aufgezeichneten Datenbestände wie Dateien, Datenbanken und Datenbankmaterial, Updates, Upgrades, Releases etc., einschließlich zugehöriger Dokumentation, Informationen und Materialien, nachfolgend als "Software" bezeichnet) zum Gegenstand haben.
- (2) Die Software wird dem Kunden auf dem maschinenlesbaren Aufzeichnungsträger überlassen, auf dem sie als Objektprogramme in ausführbarem Zustand aufgezeichnet sind. Die zur Software gehörende Anwendungsdokumentation wird dem Kunden in druckschriftlicher Form oder ebenfalls auf maschinenlesbaren Aufzeichnungsträgern überlassen. Soweit nicht ausdrücklich schriftlich vereinbart, erhält der Kunde nicht den Source Code der Software.

§ 2 Nutzungsrechte, Umfang

Bei jedweder Überlassung von Nutzungsrechten an von imc erstellter Software "Software" gelten folgende Vereinbarungen:

- (1) Grundsätzliches
 - a) Der Kunde erhält ein einfaches, nicht ausschließliches und – vorbehaltlich der Bestimmungen zur Nutzung der Software durch Dritte, Weiterveräußerung und Weitervermietung – nicht übertragbares Nutzungsrecht an der Software für eigene Zwecke. "Nutzen" umfasst die Ausführung der Programme und die Verarbeitung der Datenbestände.
 - b) Bis zur vollständigen Zahlung der jeweils fälligen Vergütung ist dem Kunden der Einsatz der Software nur widerruflich gestattet. imc kann den Einsatz solcher Leistungen, mit deren Vergütungszahlung sich der Kunde in Verzug befindet, für die Dauer des Verzuges widerrufen. Der Kunde erhält das zeitlich unbeschränkte Nutzungsrecht an urheberrechtlich geschützten Leistungen, insbesondere an der Software, nur mit vollständiger Zahlung der vereinbarten Vergütung.
 - c) Der Kunde hat geeignete Vorkehrungen zu treffen, um die Software vor dem unbefugten Zugriff Dritter zu schützen. Er wird die Originaldatenträger und die Datenträger mit den von ihm vertragsgemäß hergestellten Kopien sowie die Dokumentation an einem gesicherten Ort verwahren. Er wird seine Mitarbeiter darauf hinweisen, dass die Anfertigung von Kopien über den vertragsmäßigen Umfang hinaus unzulässig ist.
 - d) Wird das Nutzungsrecht widerrufen oder erlischt es aus einem anderen Grund, hat der Kunde die Software, die von ihm gezogenen Vervielfältigungen sowie die Dokumentation an imc herauszugeben. Falls eine körperliche Herausgabe der Software und der Vervielfältigungen aus technischen Gründen nicht möglich ist, wird der Kunde diese löschen und dies imc schriftlich bestätigen.
- (2) Vervielfältigung
 - a) Der Kunde darf die Software nur vervielfältigen, soweit dies für die vertragsgemäße Benutzung der Software erforderlich ist. Zu den notwendigen Vervielfältigungen gehören die Installation der Software vom Originaldatenträger auf die Festplatte der eingesetzten Hardware sowie das Laden der Software in den Arbeitsspeicher.
 - b) Der Kunde ist berechtigt, eine Sicherungskopie zu erstellen, wenn dies für die Sicherung künftiger Benutzung erforderlich ist. Für andere Zwecke dürfen Kopien nur nach vorheriger schriftlicher Zustimmung von imc erstellt werden.
 - c) Sonstige Vervielfältigungen, die nicht ausdrücklich gemäß den Bestimmungen dieses Vertrages erlaubt sind, sind dem Kunden nicht gestattet.

(3) Nutzung der Software durch Dritte, Weiterveräußerung und Weitervermietung

- a) Die Software darf für den vertraglich vorgesehenen Zweck, insbesondere für den Geschäftsbetrieb des Kunden genutzt werden. Sie darf ferner denjenigen zugänglich gemacht werden, die für die Benutzung der Software im Auftrag des Kunden auf diese angewiesen sind. Insbesondere darf der Kunde die Software für seine eigenen Zwecke auf Datenverarbeitungsgeräten betreiben oder betreiben lassen, die sich in den Räumen und in unmittelbarem Besitz eines dritten Unternehmens befinden (Outsourcing). Das Verbot der Mehrfachnutzung bleibt jeweils unberührt.
- b) Der Kunde darf die Software auf Dauer an Dritte veräußern oder verschenken, vorausgesetzt ihm wurde die Software zur dauerhaften Nutzung überlassen. Der Kunde darf die Software im Rahmen seiner Nutzungsdauer Dritten auch auf Zeit überlassen, sei es entgeltlich oder unentgeltlich. Das Verbot der Mehrfachnutzung bleibt jeweils unberührt. Der Kunde wird ausdrücklich darauf hingewiesen, dass die Weitergabe an Dritte nicht zulässig bzw. die Nutzung durch Dritte technisch nicht möglich ist, wenn für die Nutzung des Dritten der Erwerb einer eigenen Lizenz bzw. eine eigene Aktivierung erforderlich ist, z.B. im Fall einer sog. Runtime Lizenz.
- c) Im Fall der zulässigen Softwarenutzung durch einen Dritten hat der Kunde dafür Sorge zu tragen, dass der Dritte die Bestimmungen dieses Vertrages über die Nutzungsrechte als für sich verbindlich anerkennt. Der Kunde darf Software und Dokumentation Dritten nicht überlassen, wenn der Verdacht besteht, der Dritte werde die Bestimmungen dieses Vertrages über die Nutzungsrechte verletzen, insbesondere unerlaubte Vervielfältigungen herstellen.
- d) Vorbehaltlich der Bestimmungen in § 4 Absatz 1 und 2 oder einer abweichenden ausdrücklichen und schriftlichen Vereinbarung darf der Kunde während der Nutzung der Software durch einen Dritten die Software nicht nutzen (Verbot der Mehrfachnutzung); der Kunde übergibt bei einer Überlassung der Software an den Dritten sämtliche Softwarekopien einschließlich gegebenenfalls vorhandener Sicherheitskopien an imc oder vernichtet die nicht übergebenen Kopien.

(4) Dekompilierung

Rückübersetzungen des überlassenen Programmcodes in andere Codeformen (Dekompilierung), Entassemblierung und sonstige Arten der Rückerschließung der verschiedenen Herstellungsstufen der Software (Reverse-Engineering) sind nicht gestattet. Sollten Schnittstelleninformationen für die Herstellung der Interoperabilität eines unabhängig geschaffenen Computerprogramms erforderlich sein, so können diese gegen Erstattung eines geringen Kostenbeitrags bei imc oder einem von ihr zu benennenden Dritten angefordert werden. § 69 e UrhG bleibt von dieser Regelung unberührt.

(5) Änderungen durch imc

Führt imc Anpassungen, Änderungen bzw. Erweiterungen an der Software im Auftrag und auf Rechnung des Kunden durch, so erwirbt der Kunde an den Änderungen bzw. Erweiterungen die entsprechenden Nutzungsrechte, welche ihm nach Maßgabe dieses Vertrages an der Software zustehen.

(6) Abweichende Nutzungswünsche des Kunden

Sofern der Kunde eine Nutzung der Software wünscht, die von den in Absatz 2 bis Absatz 5 genannten Voraussetzungen abweicht, erfordert diese abweichende oder weitergehende Nutzung der Software die schriftliche Zustimmung von imc. Der Kunde wird in einem solchen Fall imc Informationen über den gewünschten Leistungsumfang, die Anwendungsgebiete etc. geben. Sofern imc daraufhin die Lizenz für diese speziell zu erstellende Applikation erteilt, sind sich die Parteien darüber einig, dass in diesem Fall eine neue Lizenzgebühr anfällt, und zwar unabhängig von der Vergütung, die bereits für das überlassene Lizenzmaterial gezahlt wurde.

§ 3 Urheberrecht, Schutz der Software

- (1) Das geistige Eigentum, insbesondere das Urheberrecht sowie alle gewerblichen Schutzrechte, und Geschäftsgeheimnisse gehen nicht auf den Kunden über, sondern verbleiben bei imc. Das Eigentum des Kunden an maschinenlesbaren Aufzeichnungsträgern, Datenspeichern und Datenverarbeitungsgeräten wird hiervon nicht berührt.
- (2) Urhebervermerke, Seriennummern sowie sonstige der Programmidentifikation oder einem Schutzrecht dienende Merkmale und Rechtsvorbehalte dürfen nicht entfernt oder verändert werden. Der Kunde ist verpflichtet, die auf der Software vorhandenen Schutzrechtsvermerke auf alle Kopien zu übernehmen. Insbesondere sind Sicherungskopien der Software ausdrücklich als solche zu kennzeichnen.

§ 4 Lizenz-Typen, Mehrfachnutzung

- (1) Im Fall einer Einzelplatzlizenz darf die Software auf einer Datenverarbeitungseinheit aktiviert und ausgeführt werden. Das Aktivieren bezeichnet den Vorgang, die Lizenz auf die Datenverarbeitungseinheit zu übertragen.

Wenn das technische Datenblatt zur Software eine zweite Aktivierung zulässt, dann darf der Kunde die Software zusätzlich auf einer zweiten Datenverarbeitungseinheit aktivieren. Die Ausführung der Software darf zu einem Zeitpunkt allerdings nicht auf beiden Datenverarbeitungseinheiten gleichzeitig erfolgen.

(2) Im Fall einer Netzwerklizenz darf die Software auf so vielen Datenverarbeitungseinheiten gleichzeitig ausgeführt werden, wie die Lizenzanzahl es vorgibt. Eine zentrale Datenverarbeitungseinheit dient dabei als Lizenzserver, auf dem auch die Aktivierung erfolgt.

Wenn das technische Datenblatt zur Software eine zweite Aktivierung zulässt, dann darf der Kunde die Software zusätzlich auf so vielen weiteren Datenverarbeitungseinheiten aktivieren und ausführen, wie die Lizenzanzahl es vorgibt. Diese weiteren Datenverarbeitungseinheiten müssen allerdings von denselben Anwendern genutzt werden, die sonst auch die Software mittels Lizenzserver betreiben.

(3) Vorbehaltlich der Bestimmungen in Absatz 1 und 2 oder einer abweichenden ausdrücklichen und schriftlichen Vereinbarung über die Netzwerknutzung ist eine Mehrfachnutzung der Software nicht gestattet.

(4) Der Kunde hat bei einem Wechsel der Datenverarbeitungseinheit die Software von der Festplatte der bisher verwendeten Hardware zu löschen.

§ 5 Software-Abonnement

Wenn es sich bei der verwendeten Software um ein Software-Abonnement handelt, dann gelten folgende zusätzliche Einschränkungen:

- (1) Das Nutzungsrecht ist zeitlich beschränkt. Der Zeitraum ist durch Beginn und Ende festgelegt. Nach dem Ende erlischt das Nutzungsrecht.
- (2) Wenn der Kunde die Software nach dem Ende des Zeitraums weiterhin nutzen möchte, muss das Abonnement verlängert werden.

§ 6 Demo-Version

Wenn es sich bei der verwendeten Software um eine kostenlose Demo-Version handelt, dann gelten folgende zusätzliche Einschränkungen:

- (1) Die Demo-Version berechtigt nur zum Test der Software. Insbesondere ist ein Produktiveinsatz nicht gestattet.
- (2) Das eingeräumte Nutzungsrecht erlischt nach Ablauf einer Zeitspanne, die der Produktbeschreibung entnommen werden kann.

§ 7 License Key

- (1) Mit der Lieferung der Software erhält der Kunde einen License Key. Mit Hilfe dieses License Keys kann der Kunde seine Software aktivieren. Ebenfalls mit Hilfe dieses License Keys kann der Kunde seinen Lizenzbestand einsehen und Updates, Upgrades bestellen.
- (2) Der License Key sollte vor dem Einblick Dritter geschützt werden, um Missbrauch auszuschließen. Sollte der Key dennoch widerrechtlich Dritten bekannt geworden sein, dann hat der Kunde imc unverzüglich telefonisch sowie auch schriftlich hierüber zu unterrichten, um den alten License Key zu sperren und einen neuen zu erhalten.

§ 8 Schlussbestimmungen

- (1) Es gilt das Recht der Bundesrepublik Deutschland unter Ausschluss der Regelungen des internationalen Privatrechts. Die Bestimmungen des UN-Übereinkommen über Verträge über den internationalen Warenkauf (CISG) finden keine Anwendung.
- (2) Erfüllungsort für sämtliche Verpflichtungen aus diesem Vertrag ist der Sitz von imc. Soweit der Kunde Kaufmann i. S. d. Handelsgesetzbuches, juristische Person des öffentlichen Rechts oder öffentlich-rechtliches Sondervermögen ist, wird als ausschließlicher Gerichtsstand für alle sich aus dem Vertragsverhältnis unmittelbar oder mittelbar ergebenden Streitigkeiten der Sitz von imc vereinbart. Dies gilt auch für Personen, die keinen allgemeinen Gerichtsstand im Inland haben, sowie für Personen, die nach Abschluss des Vertrages ihren Wohnsitz oder gewöhnlichen Aufenthaltsort ins Ausland verlegt haben oder deren Wohnsitz oder gewöhnlicher Aufenthalt im Zeitpunkt der Klageerhebung unbekannt ist. imc ist berechtigt, einen Rechtsstreit auch am gesetzlichen Gerichtsstand anhängig zu machen.
- (3) Mündliche Nebenabreden sind unwirksam. Abweichende oder ergänzende Bedingungen sowie Änderungen dieses Vertrages einschließlich dieser Schriftformklausel gelten nur, wenn sie schriftlich vereinbart und ausdrücklich als Änderung oder Ergänzung gekennzeichnet werden.
- (4) Sollten einzelne Bestimmungen dieses Vertrages unwirksam sein oder werden oder sollte der Vertrag eine Lücke enthalten, so berührt dies nicht die Gültigkeit der übrigen Bestimmungen. Anstelle der unwirksamen Bestimmung oder zur Ausfüllung einer Lücke ist eine Regelung zu vereinbaren, die, soweit rechtlich zulässig, dem am nächsten kommt, was die Vertragsparteien gewollt haben.

2 imc FAMOS Datenbank Kit

2.1 Übersicht

Dieses Kit stellt Funktionen zur Verfügung, um auf **Datenbanken** zu zugreifen. Dabei können Daten aus der Datenbank nach FAMOS, aber auch umgekehrt Daten aus FAMOS in die Datenbank übertragen werden. Das Kit gestattet den Zugriff auf:

- Oracle 10g, 11g, 12c
- MS SQL Server 2005, 2008 oder höher
- MySQL 5.5, 5.6
- Microsoft SQL Server Compact Edition 4.0
- Datenbanksysteme über ODBC

Voraussetzung ist imc FAMOS 7.0 oder eine höhere Version.

Multithreading: Alle Funktionen des Datenbank-Kits dürfen nur im Standard-Ausführungs-Thread aufgerufen werden. Ein Aufruf innerhalb eines BEGIN_PARALLEL-Blocks (also innerhalb von Sequenzfunktionen, die in einem eigenen Thread ausgeführt werden) ist nicht erlaubt.

2.2 Registrierung in imc FAMOS

Um das Datenbank-Kit in FAMOS verwenden zu können, muss es zur Benutzung registriert sein. Bei einer normal ablaufenden Installation wird diese Registrierung automatisch ausgeführt.

Sie können das Kit über die Menüfunktion "Extra-Optionen..." in FAMOS anmelden. Im Dialog "Optionen: Kits einrichten" selektieren Sie im linken Baum den Eintrag "Kits registrieren". Im rechten Teil erscheinen alle verfügbaren Kits. In dieser Liste sollte auch eine Zeile mit Datenbank-Kit [imc.Database.Kit.dll] erscheinen. Aktivieren Sie das Kontrollkästchen.

Ist dieser Eintrag nicht zu sehen, finden Sie weitere Informationen in der FAMOS- Dokumentation über die Registrierung von Kits.

2.3 Allgemeines

Die Ergebnisse von Datenbankabfragen werden in FAMOS- Datensätze und Textfelder konvertiert. Die Datensätze und Textfelder erhalten die Namen der Spalten.

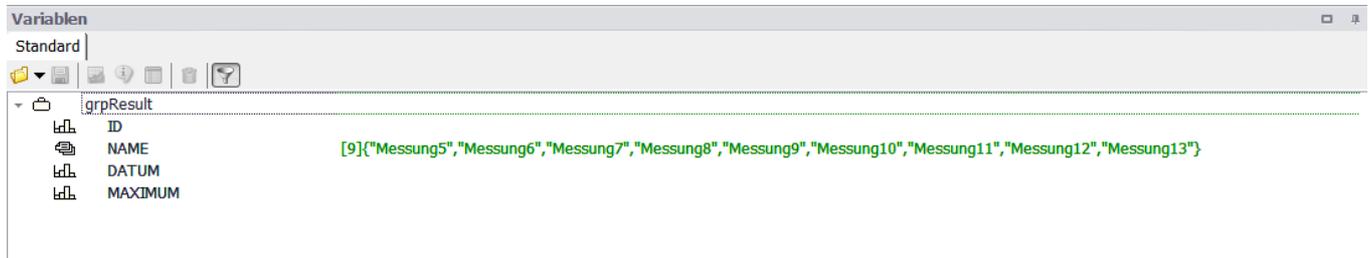
Das Beispiel soll die Umsetzung veranschaulichen. Die Tabelle Messung enthält die numerischen Spalten ID und MAXIMUM, die Textspalte NAME und die Datumsspalte DATUM.

ID	NAME	MAXIMUM	KANAL	DATUM
1	Messung1	45,3	(HugeBlob)	23.02.2016 07:58:00
2	Messung2	39,8	(HugeBlob)	23.02.2016 08:00:00
3	Messung3	46	(HugeBlob)	23.02.2016 08:03:20
4	Messung4	43,999	(HugeBlob)	23.02.2016 08:14:30
5	Messung5	34,89	(HugeBlob)	23.02.2016 08:20:20
6	Messung6	42,77	(HugeBlob)	23.02.2016 08:21:00
7	Messung7	51,01	(HugeBlob)	23.02.2016 08:30:00
8	Messung8	46,89	(HugeBlob)	23.02.2016 08:46:01
9	Messung9	47,08	(HugeBlob)	23.02.2016 08:50:20
10	Messung10	45,6	(HugeBlob)	23.02.2016 09:01:00
11	Messung11	50	(HugeBlob)	23.02.2016 09:03:45
12	Messung12	48,3	(HugeBlob)	23.02.2016 09:05:21
13	Messung13	46,987	(HugeBlob)	23.02.2016 09:06:35

Mit der Abfrage:

```
grpResult =DbSelect(ConnectID,"SELECT Id, Name, Datum, Maximum FROM
Messung WHERE Id > 4", "")
```

wird in FAMOS die Gruppe grpResult erzeugt. Die Umsetzung in Datensätze und Textfelder sieht in der Variablenliste von FAMOS folgendermaßen aus:



grpResult			
ID	Name	DATUM	MAXIMUM
5	Messung5	1140682820.0000	34.8900
6	Messung6	1140682860.0000	42.7700
7	Messung7	1140683400.0000	51.0100
8	Messung8	1140684361.0000	46.8900
9	Messung9	1140684620.0000	47.0800
10	Messung10	1140685260.0000	45.6000
11	Messung11	1140685425.0000	50.0000
12	Messung12	1140685521.0000	48.3000
13	Messung13	1140685595.0000	46.9870

Datensatz Textfeld Datensatz Datensatz

Die Spalten ID und MAXIMUM werden in je einen normalen Datensatz konvertiert. Die Spalte DATUM wird in einen Datensatz im imc Zeitformat umgewandelt. Aus der Spalte NAME ergibt sich ein Textfeld. Alle Gruppenelemente werden zu einer Gruppe zusammengefasst. Alle Gruppenelemente haben die gleiche Größe.

Beim umgekehrten Vorgang, d.h. beim Einfügen oder Aktualisieren von Daten in der Datenbanktabelle, werden die Inhalte der Gruppenelemente einer FAMOS- Gruppe in die Datentypen der Spalten der Datenbanktabelle umgewandelt und in die Datenbank geschrieben. Das Konvertieren orientiert sich immer am Datentyp der Spalte der Datenbanktabelle.

2.4 Funktionen

2.4.1 Datenbankverbindung

Funktionen zum Erzeugen und Trennen einer Datenbankverbindung

DbInitialize <small>13</small>	Definierten Anfangszustand im Kit herstellen
DbConnect <small>13</small>	Verbindung zu einem Datenbankserver erzeugen
DbDisconnect <small>16</small>	Verbindung beenden

2.4.1.1 DbInitialize

Initialisierung des Datenbank- Kit

Deklaration:

DbInitialize () -> KitVersion

Parameter:

KitVersion	Nummer der Kit Version
------------	------------------------

Beschreibung:

Die Funktion initialisiert das Datenbank- Kit. Es wird ein definierter Anfangszustand im Kit hergestellt. Eventuelle Transaktionen werden mit einem Rollback beendet. Alle Verbindungsobjekte und der interne Fehlerspeicher werden gelöscht. Die Funktion gibt die Versionsnummer des Kits zurück. Diese Rückgabe hat rein informellen Charakter.

Es wird empfohlen, diese Funktion immer am Anfang einer Sequenz aufzurufen.

Multithreading:

Alle Funktionen des Datenbank-Kits dürfen nur im Standard-Ausführungs-Thread aufgerufen werden. Ein Aufruf innerhalb eines BEGIN_PARALLEL-Blocks (also innerhalb von Sequenzfunktionen, die in einem eigenen Thread ausgeführt werden) ist nicht erlaubt.

2.4.1.2 DbConnect

Verbinden mit einem Datenbankserver

Deklaration:

DbConnect (ServerType,TxServerName,TxDatabaseName,TxUserName,TxPassword, TxExtConnectionString) -> ConnectId

Parameter:

ServerType	Typ des Datenbanksystems
	1: Microsoft SQL Server Compact Edition 4.0 mittels ADO.Net Provider
	2: Microsoft SQL Server (2005, 2008) mittels ADO.Net Provider
	3: Oracle (10g, 11g, 12c) mittels Oracle Data Provider for .NET
	4: MySQL Server (5.5, 5.6) mittels MySQL Connector/NET
	5: ODBC
TxServerName	Name des Servers
TxDatabaseName	Name der Datenbank
TxUserName	Benutzername zum Anmelden am Server
TxPassword	Passwort für den angegebenen Benutzernamen
TxExtConnectionString	Erweiterung zur Standardverbindungszeichenkette
ConnectId	Verbindungsidentifikator
	>= 1 : Gültiger Verbindungsidentifikator
	< 0 : Fehlercode

Beschreibung:

Es wird eine Verbindung zu dem angegebenen Datenbank- Server aufgebaut. Voraussetzung ist, dass ein passender ADO.Net Provider auf dem PC installiert ist.

Der Zugriff auf die Datenbanksysteme erfordert folgende ADO.Net Provider:

ServerType	ADO.Net Provider
1 (SQL Server Compact)	Microsoft SQL Server Compact Data Provider 4.0
2 (SQL Server)	.Net Framework Data Provider for SqlServer
3 (Oracle)	Oracle Data Provider for .NET (Hersteller Fa. Oracle)
4 (MySQL)	.Net Framework Data Provider for MySQL
5 (ODBC)	Ein geeigneter ODBC- Treiber muss installiert sein.

Aus den Parametern der Funktion wird eine Verbindungszeichenkette zusammengesetzt. Dann wird ein Verbindungsobjekt erstellt und es wird ein Verbindungstest durchgeführt. War auch der Verbindungstest erfolgreich, so wird ein Verbindungsidentifikator zurückgegeben. Dieser Verbindungsidentifikator repräsentiert eine Verbindung zu einem Datenbanksystem. Er ist in allen weiteren Funktionen als erster Parameter anzugeben.

Tritt ein Fehler auf, so kann mit der Funktion DbGetLastErrorText() der Fehlertext abgefragt werden. Der Aufruf DbGetLastErrorText() muss mit dem Parameter ConnectId =0 erfolgen.

- Mit TxServerName wird der Name des Datenbankservers angegeben.
- Mit TxDatabaseName wird der Name der zu verwendenden Datenbank.
- Der Parameter TxUserName ist der Benutzername zum Anmelden am Server. Sind TxUserName und TxPassword leer, so erfolgt die Anmeldung über die Windows- Authentifizierung (z.Z. nur beim Microsoft SQL Server möglich).
- Der Parameter TxPassword enthält das Passwort für den angegebenen Benutzernamen.
- Über TxExtConnectionString können weitere Elemente in der Verbindungszeichenkette gesetzt werden.

Wird die ODBC-Schnittstelle verwendet, so muss im ODBC-Manager eine Systemdatenquelle angelegt und eingerichtet sein (Auswahl des ODBC-Treibers, des Datenbankserver, der Datenbank und weiterer spezifischer Verbindungsparameter). Es sollte ein Verbindungstest im ODBC-Manager durchgeführt werden. Der Name dieser Systemdatenquelle wird als Parameter TxServerName der Funktion DbConnect() übergeben.

Aus den Funktionsparametern werden die Verbindungszeichenketten wie folgt zusammengesetzt.

ServerType	
1 (SQL Server Compact)	Data Source= TxDatabaseName ;Password= TxPassword ;Persist Security Info=False; TxExtConnectionString
2 (SQL Server)	Mit TxUserName und TxPassword Server= TxServerName ; Initial Catalog= TxDatabaseName ; Integrated Security=False;User Id= TxUserName ; Password= TxPassword ; Persist Security Info=False; TxExtConnectionString Ohne TxUserName und TxPassword Server= TxServerName ; Initial Catalog= TxDatabaseName ; Integrated Security=True; Persist Security Info=False; TxExtConnectionString
3 (Oracle)	Data Source= TxServerName ; User Id= TxUserName ; Password= TxPassword ; TxExtConnectionString
4 (MySQL)	Server= TxServerName ; Database= TxDatabaseName ; Uid= TxUserName ; Pwd= TxPassword ; TxExtConnectionString
5 (ODBC)	Dsn= TxServerName ; uid= TxUserName ;pwd= TxPassword ; TxEntConnectionsString ;

Multithreading:

Alle Funktionen des Datenbank-Kits dürfen nur im Standard-Ausführungs-Thread aufgerufen werden. Ein Aufruf innerhalb eines BEGIN_PARALLEL-Blocks (also innerhalb von Sequenzfunktionen, die in einem eigenen Thread ausgeführt werden) ist nicht erlaubt.

Beispiele:

Es wird eine Verbindung zu einer MS SQLServer Compact Edition 4.0 aufgebaut. Die Datenbank ist nicht durch ein Passwort geschützt.

```
DbInitialize()
ConnectId =DbConnect(1,"","D:\DatenbankAblage\Sample1\sample.sdf","","","")
if ConnectId < 0
    errortext=DbGetLastErrorText(0,1)
    exitsequence 1
end
```

Es wird eine Verbindung zu einem MS SQL Server mittels Windows Authentifizierung aufgebaut.

```
DbInitialize()
ConnectId=DbConnect(2,"MyPC\SQL2008EXPRESS","SampleDB","","","")
if ConnectId < 0
    errortext=DbGetLastErrorText(0,1)
    exitsequence 1
end
```

Es wird eine Verbindung zu einem Oracle-Datenbankserver ohne Verwendung der tnsnames.ora Datei hergestellt.

```
DbInitialize()
ConnectId=DbConnect(3,"(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)
    (HOST=localhost)(PORT=1522)))(CONNECT_DATA=
    (SERVER=DEDICATED)(SERVICE_NAME=orcl.berlin.imc-
    erlin.de))","","MyUsername","MyPassword","")
if ConnectId < 0
    errortext=DbGetLastErrorText(0,1)
    exitsequence 1
end
```

Es wird eine Verbindung zu einem Oracle-Datenbankserver unter Verwendung der tnsnames.ora Datei hergestellt.

```
DbInitialize()
ConnectId = DbConnect(3,"ORCL","","MyUsername"," MyPassword","")
if ConnectId < 0
    errortext=DbGetLastErrorText(0,1)
    exitsequence 1
end
```

Es wird eine Verbindung zu einem MySQL- Server aufgebaut. Dieser arbeitet über den TCP- Port 3307. Die Port- Nummer wird als Erweiterung der Verbindungszeichenkette angegeben.

```
DbInitialize()
ConnectId=DbConnect(4,"localhost","Sample_DB","MyUsername","MyPassword ",
"port=3307;")
if ConnectId < 0
    errortext=DbGetLastErrorText(0,1)
exitsequence 1
end
```

Es wird eine Verbindung zu einem MySQL- Server über ODBC aufgebaut. Im ODBC-Manager ist eine Systemdatenquelle mit dem Namen „DSN_MySQL“ eingerichtet.

```
DbInitialize()
ConnectId = DbConnect(5,"DSN_MySQL","", "MyUsername", "MyPassword", "")
if ConnectId < 0
    errortext=DbGetLastErrorText(0,1)
    EXITSEQUENCE
end
```

Verweis:

[DbDisconnect](#)¹⁶, [DbGetLastErrorText](#)³⁰, [DbGetLastErrorCode](#)³²

2.4.1.3 DbDisconnect

Verbindung zu einem Datenbankserver trennen

Deklaration:

DbDisconnect (ConnectId) -> ErrorCode

Parameter:

ConnectId	Verbindungsidentifikator
ErrorCode	= 0 : Kein Fehler
	< 0 : Fehlercode

Beschreibung:

Die Verbindung zu dem Datenbank- Server wird getrennt. Ein Fehler kann nur auftreten, wenn ein ungültiger Verbindungsidentifikator übergeben wurde.

Die Funktionen DbConnect() und DbDisconnect() sollten immer paarweise aufgerufen werden.

Beispiele:

Es wird eine Verbindung zu einem Oracle Datenbank- Server aufgebaut. Dann werden Zugriffe auf das Datenbanksystem ausgeführt. Am Ende wird die Verbindung getrennt.

```

ConnectId = DbConnect (3, "ORCL", "", "MyUsername", " MyPassword", "")
:
:
errorcode = DbDisconnect (ConnectId)

```

Verweis:

[DbConnect](#)¹³, [DbGetLastErrorText](#)³⁰, [DbGetLastErrorCode](#)³²

2.4.2 Datenzugriff

Funktionen für den Zugriff auf die Datenbank

DbSql ¹⁷	Ausführung einer SQL- Anweisung
DbSelect ¹⁸	Ausführung einer SELECT Anweisung mit Rückgabe der Daten
DbInsert ²⁰	Ausführung von INSERT Anweisungen mit Datenanbindung für mehrere Datenbankzeilen
DbUpdate1 ²³	Ausführung einer UPDATE Anweisung mit Datenübergabe
DbUpdate ²⁶	Mehrfache Ausführung einer UPDATE Anweisung mit Datenübergabe

2.4.2.1 DbSql

Ausführung einer SQL- Anweisung

Deklaration:

DbSql (ConnectId, TxSqlStatement) -> Result

Parameter:

ConnectId	Verbindungsidentifikator
TxSqlStatement	Auszuführende SQL- Anweisung Diese Anweisung wird unverändert an das Datenbanksystem übergeben.
Result	>= 0 : Anzahl der betroffenen Zeilen
	< 0 : Fehlercode

Beschreibung:

Mit dieser Funktion können SQL- Anweisungen ausgeführt werden, die keine Ergebnisse zurückliefern (UPDATE, INSERT, DELETE) (DML Kommandos). Es lassen sich aber auch DDL Kommandos wie z.B. "ALTER TABLE Messung ADD MyCol INTEGER" ausführen. Wurde ein DML- Kommando erfolgreich ausgeführt, so enthält der Rückgabewert die Anzahl der betroffenen Zeilen. Bei einem DDL- Kommando ist dieser Wert bei erfolgreicher Ausführung 0.

Die Funktion führt folgende Operationen aus:

- Die Verbindung zum Datenbankserver wird geöffnet.
- Datenbanktransaktion wird gestartet
- Die SQL- Anweisung TxSqlStatement wird ausgeführt.
- Traten Fehler auf, so wird ein Rollback ausgeführt.
- Wurde die SQL- Anweisung erfolgreich ausgeführt, so wird sie mit Commit beendet
- Die Verbindung zum Datenbankserver wird geschlossen.

Beispiele:

Alle Werte in der Spalte "Status" werden auf "Ready" gesetzt.

```
result =DbSql(ConnectId,"Update Messung set Status ='Ready'")
if result < 0
    errortext=DbGetLastErrorText(ConnectId,1)
end
```

Der Tabelle "Messung" wird die Spalte "MyCol" mit dem Datentyp Integer zugefügt.

```
result =DbSql(ConnectId,"ALTER TABLE Messung ADD MyCol Integer")
if result < 0
    errortext=DbGetLastErrorText(ConnectId,1)
end
```

Verweis:

[DbGetLastErrorText](#) ³⁰, [DbGetLastErrorCode](#) ³², [DbInsert](#) ²⁰, [DbUpdate1](#) ²³, [DbUpdate](#) ²⁶

2.4.2.2 DbSelect

Ausführung einer SELECT Anweisung mit Rückgabe der Daten

Deklaration:

DbSelect (ConnectId, TxSqlStatement, TxTimeColumn) -> GrpResult

Parameter:

ConnectId	Verbindungsidentifikator
TxSqlStatement	SQL SELECT Anweisung Diese Anweisung wird unverändert an das Datenbanksystem übergeben.
TxTimeColumn	Der Inhalt dieser Spalte wird benutzt, um den numerischen Gruppenelementen eine Zeitspur zugeben. Diese Angabe ist optional.
GrpResult	Diese Gruppe enthält das Ergebnis der Abfrage. Die ausgewählten Spalten werden in Datensätze und Textfelder konvertiert und als FAMOS- Gruppe zurückgegeben. Im Fehlerfall wird eine leere Gruppe zurückgegeben.

Beschreibung:

Mit dieser Funktion wird eine SQL SELECT- Anweisung ausgeführt. Die Funktion führt folgende Operationen aus:

- Die Verbindung zum Datenbankserver wird geöffnet.
- Die SELECT- Anweisung wird ausgeführt.
- Die Ergebnisse werden in einer Memory- Tabelle zwischengespeichert.
- Die Verbindung zum Datenbankserver wird geschlossen.
- Der Inhalt der Memory- Tabelle wird in Datensätze und Textfelder konvertiert. Diese werden zu einer FAMOS- Gruppe zusammengefasst.

Jede Spalte ergibt ein gleichnamiges Gruppenelement(Datensatz oder Textfeld). Die Anzahl der eingelesenen Zeilen bestimmt die Größe der Gruppenelemente. Im Abschnitt „[Konvertierung der Daten](#)“ ³⁴ ist eine Übersicht, wie die Datentypen der Spalten in FAMOS- Objekte konvertiert werden.

Die SQL SELECT Anweisung wird ohne Änderung an den Datenbankserver weitergegeben.

Es ist zu beachten, dass Spaltennamen gegebenenfalls quotiert werden müssen. Vergleichswerte, die aus einer Zeichenkette oder einem Datum bestehen, müssen in Hochkommata eingeschlossen werden. Mit der Funktion können auch Inhalte aus mehreren Tabellen gelesen werden. Die SELECT- Anweisung muss entsprechend zusammengestellt sein.

Enthält eine Abfrage eine oder mehrere Blob- Spalten, so darf das Ergebnis nur eine Zeile ergeben. Die Werte der Felder der Blob- Spalten ergeben je einen Datensatz. Ergibt die Abfrage mehrere Ergebniszeilen, so werden die Blob- Spalten nicht konvertiert.

War die Abfrage fehlerhaft, so wird eine leer Gruppe zurückgegeben. In diesem Fall kann über die Funktion DbGetLastErrorCode() der Fehlercode ermittelt werden.

Wird der Parameter TxTimeColumn gesetzt und entspricht er einer numerischen Spalte, so werden alle anderen numerischen Spalten in XY-Datensätze umgewandelt, mit der Zeitspalte als X-Komponente.

Über den optionalen Parameter timeStampMap = yes wird festgelegt, dass alle Spalten vom Typ Datum bzw. Zeit in ein Textfeld umgewandelt werden. Standardmäßig werden diese Spalten in einen normalen Datensatz im imc Zeitformat konvertiert.

Mit dem optionalen Parameter NumericAsDouble = yes wird festgelegt, dass alle numerischen Spalten in Datensätze mit dem Format 8 Byte reell (double) konvertiert werden. Im Standardfall erfolgt die Konvertierung wie im Abschnitt „[Konvertierung der Daten](#)“³⁴ dokumentiert.

Beispiele:

Die Abfrage

```
sqlStatement="SELECT Id,~034A B~034 ,Name FROM Messung"
grpResult = DbSelect(ConnectID, sqlStatement,"")
```

liefert eine Gruppe zurück.

In der Gruppe sind die normalen Datensätze Id und A B sowie das Textfeld Name enthalten.

Eine Besonderheit in dieser Abfrage ist die Spalte A B. Da im Namen ein Leerzeichen ist, muss diese Spalte quotiert werden. Da die Abfrage an einen Oracle- Datenbankserver gesendet wurde, muss hier das Anführungszeichen verwendet werden. In der FAMOS Sequenz ist das Ersatzzeichen ~034 zu verwenden.

Mit dieser Abfrage sollen alle Spalten der Tabelle Messung eingelesen werden.

```
grpResult = DbSelect(ConnectID,"Select * from Messung WHERE Id > 0","")
errorcode=DbGetLastErrorCode(ConnectID,0)
if errorcode < 0
    ; bringt eine Warnung, da die Spalte KANAL eine Blob-Spalte ist
    ; und das Ergebnis mehrere Zeilen enthält
    errortext=DbGetLastErrorText(ConnectID,1)
    BoxNachricht("Fehler",errortext,"S1")
end
```

Da die Spalte KANAL eine Blob- Spalte ist und im Ergebnis mehrere Zeilen vorhanden sind, erzeugt die Funktion einen Fehler. Der Inhalt der Blob- Spalte kann nicht konvertiert werden. Aus diesem Grund sind in der Ergebnisgruppe nur die Gruppenelemente ID, NAME, DATUM und MAXIMUM enthalten; das Gruppenelement KANAL fehlt. Soll der Inhalt der Blob-Spalte auch gelesen und konvertiert werden, müsste die Abfrage so geändert werden, dass nur eine Ergebniszeile pro Abfrage entsteht.

```
grpResult = DbSelect(ConnectID, "Select * from Messung WHERE Id = 1", "")
```

Verweis:

[DbGetLastErrorText](#)³⁰, [DbGetLastErrorCode](#)³², [DbOption](#)³³

2.4.2.3 DbInsert

Ausführung von INSERT Anweisungen mit Datenanbindung für mehrere Datenbankzeilen.

Deklaration:

DbInsert (ConnectId, TxTableName, GrpInsertValues) -> Result

Parameter:

ConnectId	Verbindungsidentifikator
TxTableName	Name der Datenbanktabelle, in die Werte eingefügt werden sollen
GrpInsertValues	Die FAMOS- Gruppe enthält Daten, die zum Übertragen in die Datenbank benutzt werden. Jedes Sample der Gruppenelemente führt zum INSERT einer neuen Zeile in der Tabelle
Result	>= 0 : Anzahl der eingefügten Zeilen
	< 0 : Fehlercode

Beschreibung:

Mit dieser Funktion können Daten aus FAMOS- Objekten in eine Datenbanktabelle eingefügt werden. Die Namen der Datensätze und Textfelder in der Gruppe müssen in der Datenbanktabelle als Spalten vorhanden sein. Es werden alle Elemente, die in der Gruppe sind in die Tabelle eingefügt. Die Datensätze und Textfelder der Gruppe müssen die gleiche Größe haben. Die Größe bestimmt die Anzahl der eingefügten Tabellenzeilen.

Wenn eine Blob- Spalte dabei ist, so darf die Größe der Gruppenelemente nur 1 sein. Der Name des Datensatzes, der der Blob- Spalte entspricht, wird in das eine Feld der Tabelle eingefügt. Haben die Gruppenelemente eine Größe > 1, so werden nur die Daten mit dem ersten Index eingefügt.

Die Funktion führt folgende Operationen aus:

- Von der angegebenen Tabelle wird eine leere Memory- Tabelle angelegt.
- Die Datensätze und Textfelder der FAMOS- Gruppe werden gegen die Memory- Tabelle geprüft.
Existiert für jedes Gruppenelement eine gleichnamige Spalte ?
Haben alle Gruppenelemente die gleiche Anzahl von Werten ?
Haben alle Datensätze der FAMOS-Gruppe den Datentyp „normal“ ?
- Die Daten aus den Elementen der FAMOS- Gruppe werden konvertiert und in der Memory- Tabelle zwischengespeichert. Die Konvertierung richtet sich immer nach dem Datentyp der Spalte der Datenbanktabelle.
- Die Verbindung zum Datenbankserver wird geöffnet.
- Eine Datenbanktransaktion wird gestartet.
- Aus dem Tabellennamen und den Namen der Gruppenelemente wird eine INSERT- Anweisung aufgebaut. Alle Spaltennamen werden quotiert.

- Pro Zeile in der Memory-Tabelle wird ein INSERT-Kommando mit den entsprechenden Werten an den Datenbankserver geschickt.
- Traten im INSERT-Verlauf Fehler auf, so wird ein Rollback ausgeführt.
- Wurden alle INSERT-Kommandos erfolgreich ausgeführt, so wird ein Commit ausgeführt.
- Die Verbindung zum Datenbankserver wird geschlossen.

Beispiele:

Es wird eine neue Zeile mit einer Blob-Spalte erzeugt.

```
; --- Einfügen eines Kanals als Blob -----
; --- Wichtig! Es darf nur eine Zeile geladen werden -----
grpResult = DbSelect(ConnectID,"Select * from Messung WHERE Id = 1","")
errorcode = DbGetLastErrorCode(ConnectID,0)
if errorcode < 0
    errortext = DbGetLastErrorText(ConnectID,1)
    BoxNachricht("Fehler",errortext,"S1")
end
; --- Neuen ID Wert für das Einfügen bestimmen -----
grpMax = DbSelect(ConnectID,"Select Max(Id) AS MaxID from Messung ","")
maxid = grpMax:MaxID[1]
grpResult:ID[1]=maxid+1
; --- Kanal Sintest1 laden -----
FileLoad("Sintest1.dat","",",0)
grpResult:KANAL=sintest1
; --- Triggerzeit als Datum speichern -----
grpResult:Datum[1]=Zeit?(grpResult:KANAL)
grpResult:Name[1] ="Sintest1"
grpResult:Maximum[1]=Max(sintest1)
; --- Kanal wird als Blob in die Tabelle eingefügt -----
; --- Wichtig! Die Gruppenelemente dürfen nur die Größe 1 haben -
Result = DbInsert(ConnectID,"Messung",grpResult)
if result < 0
    errortext = DbGetLastErrorText(ConnectID,1)
    BoxNachricht("Fehler",errortext,"S1")
end
```

Mehrere Zeilen werden aus der Tabelle gelesen, verändert und als neue Zeilen eingefügt.

```

grpResult =DbSelect(ConnectID,"Select Id, Name, Datum, Maximum from Messung
                    WHERE Id > 5 And Name like 'Mess%' ", "")
; --- Neuen ID Wert für das Einfügen bestimmen -----
grpMax = DbSelect(ConnectID,"Select Max(Id) AS MaxID from Messung ", "")
; --- Neue ID Werte für das Einfügen setzen -----
maxid = grpMax:MaxID[1]+1
FOREACH SAMPLE s in grpResult:ID
    s=s+maxid
END
; --- Neue Namen für das Einfügen setzen -----
FOREACH ELEMENT s in grpResult:NAME
    s=s+"_Neu"
END
; --- Neues Datum für das Einfügen setzen -----
FOREACH SAMPLE s in grpResult:Datum
    s=s+ 86400
END
; --- Neues Maximum für das Einfügen setzen -----
FOREACH SAMPLE s in grpResult:Maximum
    s=0
END
; --- Transaktion beginnen ( ist eigentlich nicht notwendig, ----
; --- dient zum Test der Funktionen DbBeginTransaction() und ---
; --- DbEndTransaction() -----
DbBeginTransaction(ConnectID)
; --- Gruppe in die Tabelle Messung einfügen -----
result=DbInsert(ConnectID,"Messung",grpResult)
if result < 0
    errortext=DbGetLastErrorText(ConnectID,1)
    BoxNachricht("Fehler",errortext,"S1")
    ; --- Transaktion mit einem Rollback abschließen -----
    DbEndTransaction(ConnectId,0)
Else
    ; --- Transaktion mit einem Commit abschließen -----
    DbEndTransaction(ConnectId,1)
End

```

Verweis:

[DbGetLastErrorText](#) ^[30], [DbGetLastErrorCode](#) ^[32], [DbBeginTransaction](#) ^[28], [DbEndTransaction](#) ^[30]

2.4.2.4 DbUpdate1

Ausführung einer SQL- Update- Anweisung

Deklaration:

DbUpdate1 (ConnectId, TxSqlStatement, GrpUpdateValues, SampleIndex) -> Result

Parameter:

ConnectId	Verbindungsidentifikator
TxSqlStatement	SQL UPDATE Anweisung In der Anweisung werden die zu setzenden Werte nicht angegeben, sondern nur Platzhalter vom Typ ?Gruppenelementname.
GrpUpdateValues	Enthält Daten zum Aktualisieren und für die WHERE-Bedingung Die FAMOS- Gruppe enthält Daten, die zum Aktualisieren und für die WHERE-Bedingung benutzt werden. Es werden nur die Daten der Gruppenelemente verwendet, die durch den SampleIndex adressiert sind.
SampleIndex	Nur mit den Werten aus den Gruppenelementen, die durch den SampleIndex adressiert werden, erfolgt eine Aktualisierung der Datenbankwerte (1...)
Result	>= 0 : Anzahl der aktualisierten Zeilen
	< 0 : Fehlercode

Beschreibung:

Mit dieser Funktion wird eine UPDATE-Anweisung ausgeführt. Die UPDATE-Anweisung muss komplett als TxSqlStatement angegeben sein. Als Platzhalter für die zu ersetzenden Werte aus den Gruppenelementen ist das Fragezeichen **?+Name des Gruppenelements** zu verwenden. Die Platzhalter im SET-Teil und in der WHERE-Bedingung werden durch die Daten aus der Gruppe GrpUpdateValues ersetzt.

Für die Update-Anweisung werden nur die Daten aus den Gruppenelementen verwendet, die durch den SampleIndex adressiert sind. In der Gruppe können mehr Gruppenelemente vorhanden sein, als für die Update-Anweisung benötigt werden. Es werden nur die Gruppenelemente benutzt, die in der Update-Anweisung angegeben sind.

Die Funktion führt folgende Operationen aus:

- Die SQL UPDATE -Anweisung wird auf das Vorhandensein der Schlüsselwörter UPDATE und SET geprüft.
- Der Tabellename wird aus der UPDATE- Anweisung ermittelt. Von der angegebenen Tabelle wird eine leere Memory- Tabelle angelegt.
- Im SET- Teil werden die zu aktualisierenden Spalten bestimmt. Der Platzhalter **?Gruppenelementname** wird durch einen benannten Parameter des Datenbanksystems ersetzt. Es muss ein gleichnamiges Gruppenelement in der Gruppe GrpUpdateValues vorhanden sein.
- In der WHERE- Bedingung werden die Platzhalter **?Gruppenelementname** durch benannte Parameter des Datenbanksystems ersetzt. Es muss ein gleichnamiges Gruppenelement in der Gruppe GrpUpdateValues vorhanden sein.
- Die Daten, die über den SampleIndex adressiert sind, werden in die entsprechenden Datentypen der Memory- Tabelle konvertiert.
- Die Verbindung zum Datenbankserver wird geöffnet.
- Eine Datenbanktransaktion wird gestartet.
- Es wird die UPDATE- Anweisung ausgeführt.
- Traten Fehler auf, so wird ein Rollback ausgeführt.
- Wurde das UPDATE- Kommandos erfolgreich ausgeführt, so wird ein Commit ausgeführt.
- Die Verbindung zum Datenbankserver wird geschlossen.

Beispiele:

Das MAXIMUM der Zeile mit dem ID-Wert 11 soll auf 0 gesetzt werden.

```
grpResult =DbSelect(ConnectID, "Select Id,Name,Datum,Maximum from Messung",  
    "")  
grpResult:Maximum[11]=0;  
updatestatement="update Messung set Maximum=?Maximum where Id =?Id "  
result=DbUpdate1(ConnectId,updatestatement,grpResult,11)  
if result < 0  
    errortext=DbGetLastErrorText(ConnectID,1)  
    BoxNachricht("Fehler",errortext,"S1")  
End
```

Zuerst werden die Werte der Spalten ID, NAME, DATUM und MAXIMUM eingelesen.

Die folgende Abbildung zeigt die Bestandteile FAMOS- Gruppe. Der 11. Wert vom Datensatz MAXIMUM wird auf 0 gesetzt.

Die UPDATE- Anweisung "Update Messung Set Maximum=?Maximum Where Id =?ID" enthält 2 Platzhalter. Der erste Platzhalter ist im SET-Teil und der zweite in der WHERE- Bedingung.

Die Platzhalter werden durch die konkreten Werte, die vom SampleIndex = 11 gelesen werden, ersetzt. Die Spalten NAME und Datum sind auch in der Gruppe. Sie werden aber nicht berücksichtigt, da nur die Spalten MAXIMUM und ID in der UPDATE-Anweisung mit einem Platzhalter versehen sind. In diesem Fall wird an den Datenbankserver das UPDATE- Kommando

"Update Messung Set Maximum=0 Where Id =11"

geschickt. Das Ergebnis der Funktion ist 1, da die WHERE-Bedingung nur eine Zeile in der Datenbanktabelle zulässt.

Sample Index	grpResult			
	ID	Name	DATUM	MAXIMUM
1	1	Messung1	1140681480.0000	45.3000
2	2	Messung2	1140681600.0000	39.8000
3	3	Messung3	1140681800.0000	46.0000
4	4	Messung4	1140682470.0000	43.9990
5	5	Messung5	1140682820.0000	34.8900
6	6	Messung6	1140682860.0000	42.7700
7	7	Messung7	1140683400.0000	51.0100
8	8	Messung8	1140684361.0000	46.8900
9	9	Messung9	1140684620.0000	47.0800
10	10	Messung10	1140685260.0000	45.6000
-> 11	11	Messung11	1140685425.0000	50.0000 -> 0
12	12	Messung12	1140685521.0000	48.3000
13	13	Messung13	1140685595.0000	46.9870

↑

↑

Das Gruppenelement wird in der WHERE-Bedingung verwendet.	Das Gruppenelement wird in der UPDATE-Klausel verwendet
---	---

Im nächsten Beispiel sollen die Werte von MAXIMUM auf 1000 gesetzt werden, wo der Name mit 1x endet.

```
grpResult =DbSelect (ConnectID, "Select Id,Name,Datum,Maximum from Messung",
                    "")
grpResult:Maximum[1]=1000;
updatestatement="update Messung set Maximum=?Maximum where Name LIKE '%1_' "
result=DbUpdate1 (ConnectId,updatestatement,grpResult, 1)
if result < 0
    errortext=DbGetLastErrorText (ConnectID,1)
    BoxNachricht ("Fehler",errortext,"S1")
end
```

Nach dem Einlesen der Spalten wird der Wert vom Maximum mit dem SampleIndex = 1 auf 1000 gesetzt.

In der UPDATE- Anweisung ist nur ein Platzhalter im SET-Teil. Die WHERE- Bedingung "Name LIKE '%1_'" filtert 4 Zeilen, die aktualisiert werden. Das Ergebnis der Funktion ist 4.

Mit den folgenden Sequenzschritten wird in einer Blob-Spalte ein Kanal aktualisiert.

```
grpResult =DbSelect (ConnectID, "Select Id, Name, Datum, Maximum, Kanal from
Messung where ID=10 ", "")
; --- Kanal Sintest1 laden -----
FileLoad("Sintest1.dat", "", 0)
; --- Der Datensatz sintest1 wird dem Gruppenelement Kanal zugewiesen-
grpResult:KANAL=sintest1
updatestatement="update Messung set Kanal=?Kanal where Id =?Id"
result=DbUpdate1 (ConnectId, updatestatement, grpResult, 1)
if result < 0
    errortext=DbGetLastErrorText (ConnectID, 1)
    BoxNachricht ("Fehler", errortext, "S1")
end
```

Die Zeile mit dem ID = 10 wird aus der Datenbanktabelle eingelesen. Es wird der Kanal sintest1 gelesen und dem Gruppenelement KANAL zugewiesen. Durch das Ausführen von DbUpdate1() wird der ganze Kanal dem Blob-Feld KANAL in der Tabellenzeile mit ID=10 zugewiesen. Das Ergebnis der Funktion ist 1.

Verweis

[DbSelect](#)^[18], [DbUpdate](#)^[26], [DbGetLastErrorText](#)^[30], [DbGetLastErrorCode](#)^[32], [DbBeginTransaction](#)^[28],
[DbEndTransaction](#)^[30]

2.4.2.5 DbUpdate

Mehrfache Ausführung einer SQL- Update- Anweisung

Deklaration:

DbUpdate (ConnectId, TxSqlStatement, GrpUpdateValues) -> Result

Parameter:

ConnectId	Verbindungsidentifikator
TxSqlStatement	SQL UPDATE Anweisung In der Anweisung werden die zu setzenden Werte nicht angegeben, sondern nur Platzhalter ?Gruppenelementname.
GrpUpdateValues	Enthält Daten zum Aktualisieren und für die WHERE- Bedingung Die FAMOS- Gruppe enthält Daten, die zum Aktualisieren und für die WHERE- Bedingung benutzt werden.
Result	>= 0 : Anzahl der aktualisierten Zeilen
	< 0 : Fehlercode

Beschreibung:

Mit dieser Funktion wird die UPDATE-Anweisung pro Index der übergebenen Datensätze ausgeführt. Die UPDATE-Anweisung muss komplett als TxSqlStatement angegeben sein. Als Platzhalter für die zu ersetzenden Werte aus den Gruppenelementen ist das Fragezeichen **?+Gruppenelementname** zu verwenden.

Ist in der UPDATE- Anweisung eine Blob- Spalte, so wird nur die 1. Zeile in der Datenbanktabelle aktualisiert.

Die Funktion führt folgende Operationen aus:

- Die SQL UPDATE -Anweisung wird auf das Vorhandensein der Schlüsselwörter UPDATE und SET geprüft.

- Der Tabellename wird aus der UPDATE- Anweisung ermittelt. Von der angegebenen Tabelle wird eine leere Memory- Tabelle angelegt.
- Im SET- Teil werden die zu aktualisierenden Spalten bestimmt. Der Platzhalter **?Gruppenelementname** wird durch einen benannten Parameter des Datenbanksystems ersetzt. Es muss ein gleichnamiges Gruppenelement in der Gruppe GrpUpdateValues vorhanden sein.
- In der WHERE- Bedingung werden die Platzhalter **?Gruppenelementname** durch benannte Parameter des Datenbanksystems ersetzt. Es muss ein gleichnamiges Gruppenelement in der Gruppe GrpUpdateValues vorhanden sein.
- Die Daten in der Gruppe GrpUpdateValues werden in die entsprechenden Datentypen der Memory- Tabelle konvertiert. Alle Gruppenelemente müssen die gleiche Anzahl von Werten haben
- Die Verbindung zum Datenbankserver wird geöffnet.
- Eine Datenbanktransaktion wird gestartet.
- Pro Zeile der Memory- Tabelle wird die UPDATE- Anweisung ausgeführt.
- Traten Fehler auf, so wird ein Rollback ausgeführt.
- Wurde das UPDATE- Kommando erfolgreich ausgeführt, so wird ein Commit ausgeführt.
- Die Verbindung zum Datenbankserver wird geschlossen.

Beispiele:

Die Spalten ID, NAME und DATUM werden eingelesen. Die Gruppenelemente haben eine Größe von 13. Jetzt werden die einzelnen Namen verändert. Die veränderte Gruppe aktualisiert abschließend die Datenbanktabelle.

```
grpResult =DbSelect(ConnectID,"Select Id, Name,Datum from Messung","")
; --- Neue Namen für das Aktualisieren bilden -----
for i= 1 to lang?(grpResult:ID)  Step 1
    messung=grpResult:Name[i] + "_" + TForm(grpResult:ID[i],"")
    grpResult:Name[i]=messung
end
```

```
updatestatement="update Messung set Name=?Name where Id =?Id "
result=DbUpdate(ConnectId,updatestatement,grpResult)
if result < 0
    errortext=DbGetLastErrorText(ConnectID,1)
    BoxNachricht("Fehler",errortext,"S1")
end
```

Die UPDATE-Anweisung wird 13 mal ausgeführt. Bei jedem UPDATE werden die Platzhalter durch die entsprechenden Daten aus den Gruppenelementen ersetzt. Das Ergebnis der Funktion ist 13. In der Abbildung ist der Tabelleninhalt vor und nach dem Aktualisieren zu sehen. Die Spalte DATUM wird zwar mit eingelesen, da sie aber in der UPDATE-Anweisung nicht erscheint, wird sie nicht berücksichtigt.

Tabelle vorher

ID	Name
1	Messung1
2	Messung2
3	Messung3
4	Messung4
5	Messung5
6	Messung6
7	Messung7
8	Messung8
9	Messung9
10	Messung10
11	Messung11
12	Messung12
13	Messung13

Tabelle nachher

ID	Name
1	Messung1_1
2	Messung2_2
3	Messung3_3
4	Messung4_4
5	Messung5_5
6	Messung6_6
7	Messung7_7
8	Messung8_8
9	Messung9_9
10	Messung10_10
11	Messung11_11
12	Messung12_12
13	Messung13_13

Verweise

[DbSelect](#)^[18], [DbUpdate1](#)^[23], [DbGetLastErrorText](#)^[30], [DbGetLastErrorCode](#)^[32],
[DbBeginTransaction](#)^[28], [DbEndTransaction](#)^[30]

2.4.3 Transaktionen

Funktionen zur Steuerung von Transaktionen

DbBeginTransaction ^[28]	Transaktion beginnen
DbEndTransaction ^[30]	Transaktion beenden

2.4.3.1 DbBeginTransaction

Eine Datenbanktransaktion wird gestartet.

Deklaration:

DbBeginTransaction (ConnectId) -> ErrorCode

Parameter:

ConnectId	Verbindungsidentifikator
ErrorCode	= 0 : Kein Fehler
	< 0 : Fehlercode

Beschreibung:

Mit dieser Funktion wird eine Transaktion begonnen. Diese Funktion kann verwendet werden, wenn mehrere Aufgaben in der Sequenz miteinander verbunden werden sollen, damit sie als eine einzelne Verarbeitungseinheit ausgeführt werden können.

Es sollen beispielsweise Daten in die Tabelle A und gleichzeitig korrespondierende Daten in Tabelle B eingefügt werden. Gibt es beim Einfügen in die Tabelle B einen Fehler, so soll das Einfügen in Tabelle A rückgängig gemacht werden. Das kann mit dem paarweisen Einsatz von `DbBeginTransaction()` und `DbEndTransaction()` sichergestellt werden.

Diese Funktionen müssen nicht verwendet werden, wenn nur ein einzelner Aufruf von `DbSql()`, `DbInsert()`, `DbUpdate()` oder `DbUpdate1()` erfolgt. Diese Funktionen haben eine interne Transaktionssteuerung. Sollen beispielsweise die Elemente einer Gruppe mit dem `DbInsert()` eingefügt werden, so werden entweder alle Daten der Elemente eingefügt (es wird intern ein Commit ausgeführt) oder wenn ein Fehler auftrat werden alle Änderungen verworfen (es wird intern ein Rollback ausgeführt).

Wird die Funktion `DbBeginTransaction()` aufgerufen, so wird die interne Transaktionssteuerung außer Kraft gesetzt. Erst nach dem Aufruf von `DbEndTransaction()` wird die interne Transaktionssteuerung wieder aktiv.

Eine Schachtelung von Aufrufen von `DbBeginTransaction()` wird nicht unterstützt.

Die Funktion führt folgende Operationen aus:

- Die Verbindung zum Datenbankserver wird geöffnet.
- Datenbanktransaktion wird begonnen.
- Die Verbindung zum Datenbankserver bleibt geöffnet.

Benutzen Sie `DbBeginTransaction()` und `DbEndTransaction()` immer paarweise.

Trifft die Funktion `DbDisconnect()` auf eine offene Transaktion, so wird diese mit einem Rollback abgeschlossen.

Beispiele:

In die Tabelle "Benutzer" wird die FAMOS- Gruppe `grpInsert1` und in die Tabelle "Gruppe" die FAMOS-Gruppe `grpInsert2` eingefügt. Dazu wird eine Transaktion gestartet. Wurden beide Funktionen ohne Fehler beendet, so wird die Transaktion mit einem Commit beendet. Im Fehlerfall erfolgt ein Rollback.

```
errorcode=DbBeginTransaction (ConnectID)
if errorcode < 0
    errortext=DbGetLastErrorText (ConnectID,1)
end
result1=DbInsert (ConnectID, "Benutzer", grpInsert1)
result2=DbInsert (ConnectID, "Gruppe", grpInsert2)

commit=1
if result1 < 0 or result2 < 0
    errortext=DbGetLastErrorText (ConnectID,1)
    commit=0;
end
errorcode = DbEndtransaction (ConnectID, commit)
if errorcode < 0
    errortext=DbGetLastErrorText (ConnectID,1)
end
```

Verweis:

[DbEndTransaction](#) ^[30], [DbGetLastErrorText](#) ^[30], [DbGetLastErrorCode](#) ^[32]

2.4.3.2 DbEndTransaction

Eine Datenbanktransaktion wird festgeschrieben oder zurückgesetzt.

Deklaration:

DbEndTransaction (ConnectId, Commit) -> ErrorCode

Parameter:

ConnectId	Verbindungsidentifikator
Commit	= 0 : Die Transaktion wird zurückgesetzt. Alle Änderungen werden verworfen (Rollback)
	<>0 : Die Transaktion wird festgeschrieben. Alle Änderungen werden dauerhaft gespeichert. (Commit)
ErrorCode	= 0 : Kein Fehler
	< 0 : Fehlercode

Beschreibung:

Mit dieser Funktion wird eine Transaktion beendet. Die Anwendung ist nur im Zusammenspiel mit der Funktion DbBeginTransaction() erlaubt.

Benutzen Sie DbBeginTransaction() und DbEndTransaction() immer paarweise.

Die Funktion führt folgende Operationen aus:

- Die Transaktion wird entsprechend dem Parameter "Commit" festgeschrieben oder zurückgesetzt.
- Die Verbindung zum Datenbankserver wird geschlossen.

Verweis:

[DbBeginTransaction](#) ^[28], [DbGetLastErrorText](#) ^[30], [DbGetLastErrorCode](#) ^[32]

2.4.4 Verschiedenes

Funktionen zur Fehleranalyse und für Optionen

DbGetLastErrorText ^[30]	Letzten Fehlertext ermitteln
DbGetLastErrorCode ^[32]	Letzten Fehlercode ermitteln
DbOption ^[33]	Optionale Parameter setzen

2.4.4.1 DbGetLastErrorText

Ermittelt den letzten Fehlertext

Deklaration:

DbGetLastErrorText (ConnectId, ClearError) -> TxError

Parameter:

ConnectId	Verbindungsidentifikator
ClearError	0 : Fehler bleibt im Fehlerspeicher
	1 : Fehlerspeicher anschließend löschen
TxError	Fehlertext

Beschreibung:

Mit dieser Funktion kann der Text zu dem letzten aufgetretenen Fehler gelesen werden. Wird für den Parameter "ClearError" eine 1 angegeben, so wird anschließend der Fehlerspeicher gelöscht. Jede Verbindung hat einen eigenen Fehlerspeicher. Aus diesem Grund ist der Parameter "ConnectId" anzugeben.

Schlägt die Funktion DbConnect() fehlt, so ist beim Aufruf von DbGetLastError() der Parameter ConnectId =0 anzugeben (DbGetLastError(0,1)).

Im Abschnitt [Fehlercodes](#)³⁷ sind die Fehler beschrieben.

Verweis:

[DbGetLastErrorCode](#)³²

2.4.4.2 DbGetLastErrorCode

Ermittelt die letzte Fehlernummer

Deklaration:

DbGetLastErrorCode (ConnectId, ClearError) -> ErrorCode

Parameter:

ConnectId	Verbindungsidentifikator
ClearError	0 : Fehler bleibt im Fehlerspeicher 1 : Fehlerspeicher anschließend löschen
ErrorCode	Fehlernummer

Beschreibung:

Mit dieser Funktion kann die Fehlernummer zu dem letzten aufgetretenen Fehler gelesen werden. Wird für den Parameter "ClearError" eine 1 angegeben, so wird anschließend der Fehlerspeicher gelöscht. Jede Verbindung hat einen eigenen Fehlerspeicher. Aus diesem Grund ist der Parameter "ConnectId" anzugeben.

Die meisten Kit- Funktionen liefern im Fall eines Fehlers im Ergebnis die Fehlernummer zurück. Fehlernummern sind immer negative Zahlen.

Im Abschnitt [Fehlercodes](#)³⁷ sind die Fehler beschrieben.

Verweis:

[DbGetLastErrorText](#)³⁰

2.4.4.3 DbOption

Setzen von optionalen Parametern

Deklaration:

DbOption (ConnectId, TxParametername, TxParameterValue) -> ErrorCode

Parameter:

ConnectId	Verbindungsidentifikator
TxParametername	Name des optionalen Parameters
TxParameterValue	Werte des optionalen Parameters
ErrorCode	=0 : Kein Fehler
	<0 : Fehlercode

Folgende Parameter sind vereinbart:

Parametername	Wert	Bemerkung
"TimeStampMap"	"yes" oder "no"	Beeinflusst die Funktion DbSelect() Ist der Parameter auf "yes" gesetzt, so werden die Inhalte von DateTime- Spalten in ein Textfeld umgewandelt. Bei einem "no" werden diese Spalten in einen normalen Datensatz konvertiert. Standardwert: "no"
"NumericAsDouble"	"yes" oder "no"	Beeinflusst die Funktion DbSelect() Ist der Parameter auf "yes" gesetzt, so werden die Inhalte von numerischen Spalten immer in einen normalen Datensatz mit dem Datenformat 8 Byte reell konvertiert. Ist der Wert "no" erfolgt die Konvertierung wie im Abschnitt „ Konvertierung der Daten “ angegeben. Standardwert: "no"
"CommandTimeOut"	"xx" in Sekunden	Beeinflusst die Funktionen DbSql(), DbSelect(), DbInsert(), DbUpdate1() und DbUpdate() Über diesen Parameter wird die Zeit in Sekunden festgelegt, die gewartet werden soll, bis der Versuch einer Befehlsausführung beendet und ein Fehler generiert wird. Beim Microsoft SQL Server Compact Edition 4.0 hat dieser Parameter keine Bedeutung. Der Standardwert sind 30 Sekunden.

Beschreibung:

Mit dieser Funktion können optionale Parameter pro Verbindung gesetzt werden.

Der Name des optionalen Parameters muss einem vereinbarten Namen entsprechen.

Verweis:

[DbSelect](#)^[18], [DbSql](#)^[17], [DbInsert](#)^[20], [DbUpdate1](#)^[23], [DbUpdate](#)^[26], [DbGetLastErrorText](#)^[30],
[DbGetLastErrorCode](#)^[32]

2.5 Konvertierung der Daten

Beim Lesen von Daten aus einer Tabelle werden die Werte entsprechend dem Datentyp der Spalten in Datensätze bzw. Textfelder konvertiert.

Datentyp der Memory-Tabelle	Wertebereich	Option	FAMOS-Objekt	FAMOS- Datenformat
Boolean	true, false	1	Normaler Datensatz	1 Byte ganzzahlig
Byte	0...255	1	Normaler Datensatz	1 Byte ganzzahlig ohne Vorzeichen
Char			Textfeld	
DateTime		2	Normaler Datensatz	8 Byte reell
			Textfeld	
DateTime Offset		2	Normaler Datensatz	8 Byte reell
			Textfeld	
Decimal	- 792281625142643375935439 50335m 792281625142643375935439 50335m		Normaler Datensatz	8 Byte reell
Double	-1,79769e+308 ... 1,79769e+308		Normaler Datensatz	8 Byte reell
Guid			Textfeld	
Int16	-32768 ... 32767	1	Normaler Datensatz	2 Byte ganzzahlig
Int32	-2147483648 ... 2147483647	1	Normaler Datensatz	4 Byte ganzzahlig
Int64	-9223372036854775808 ... 9223372036854775807	1	Normaler Datensatz	8 Byte ganzzahlig
SByte	-128 ... 127	1	Normaler Datensatz	1 Byte ganzzahlig
Single	-3,40282e+038f ... 3,40282e+038f	1		4 Byte reell
String			Textfeld	
TimeSpan		2	Normaler Datensatz	8 Byte reell
			Textfeld	
UInt16	0 ... 65535	1	Normaler Datensatz	2 Byte ganzzahlig ohne Vorzeichen
UInt32	0 ... 4294967295	1	Normaler Datensatz	4 Byte ganzzahlig ohne Vorzeichen
UInt64	0... 18446744073709551615	1		8 Byte ganzzahlig ohne Vorzeichen
Byte[]			Normaler Datensatz	1 Byte ganzzahlig ohne Vorzeichen

Liegt ein Wert der Datentypen Decimal, Double oder Single außerhalb der Grenzen des FAMOS –Wertebereichs, so wird dieser Wert auf den Grenzwert gesetzt. (-10^{35} ... 10^{35}).

Wird eine Datum-/Zeitspalte in einen Datensatz im imc Zeitformat konvertiert und liegt eine Zeitangabe vor 1980, so wird diese auf den FAMOS- Grenzwert 1.1.1980 gesetzt.

Option 1

Standardmäßig wird der Datentyp der Spalte in das angegebene FAMOS- Datenformat konvertiert. Wird der optionalen Parameter "NumericAsDouble=yes" gesetzt, so werden diese Spalten in das FAMOS- Format 8 Byte reell konvertiert.

Option 2

Standardmäßig wird die Spalte in einen normalen Datensatz konvertiert. Wird der optionale Parameter "TimeStampMap=yes" gesetzt, so werden diese Spalten in ein Textfeld umgewandelt.



Hinweis

Die folgenden Datentypen des Microsoft SQL Server werden nicht unterstützt:

- SqlGeography
- SqlGeometry
- SqlHierarchyId
- SQLVariant

2.6 Behandlung von Nullwerten

Nullwerte aus den Datenbanktabellen können nicht in den FAMOS- Objekten (Datensätze und Textfelder) gespeichert werden.

Werden Nullwerte in Textfelder konvertiert, so ist der Wert immer eine leere Zeichenkette.

Bei der Umwandlung von Nullwerten von numerischen Spalten in FAMOS- Datensätze gelten folgende Ersatzwerte:

Datyp der Memory-Tabelle	Wertebereich	FAMOS-Datenformat im Datensatz	Ersatzwert für den Nullwert
Boolean	True, false	Signed1	-128
Byte	0...255	Unsigned1	0
DateTime		Double	0.0
Decimal	-79228162514264337593543950335m 79228162514264337593543950335m	Double	1e35
Double	-1,79769e+308 ... 1,79769e+308	Double	1e35
Int16	-32768 ... 32767	Signed2	-32768
Int32	-2147483648 ... 2147483647	Signed4	-2147483648
Int64	-9223372036854775808 ... 9223372036854775807	Signed8	-9223372036854775808
SByte	-128 ... 127	Signed1	-128
Single	-3,40282e+038f ... 3,40282e+038f	Float	1e35
TimeSpan		Double	1e35
UInt16	0 ... 65535	Unsigned2	0
UInt32	0 ... 4294967295	Unsigned4	0
UInt64	0... 18446744073709551615	Unsigned8	0

2.7 Fehlercodes

Die Fehlernummern sind immer negative Zahlen.

Fehlernummer	Beschreibung
-1	Für einen Parameter wurde kein Wert angegeben. Z.B. eine leere SQL- Anweisung oder ein fehlender Tabellenname bzw. Servername.
-2	Die Funktion DbConnect() liefert einen Verbindungsidentifikator. Alle anderen Funktionen erwarten einen gültigen Verbindungsidentifikator und prüfen diesen, ob eine Verbindung existiert. Ist das nicht der Fall, so erscheint diese Fehlernummer.
-3	Provider Fehler: Fehlertext Dieser Fehler kann beim Öffnen der Verbindung zu dem Datenbankserver erscheinen. Der Provider wirft eine Ausnahme. Im Fehlertext ist die Ursache dieser Ausnahme zu sehen.
-4	DBMS Fehler: Fehlertext Ist eine SQL- Anweisung (SELECT, UPDATE ...) fehlerhaft, so meldet der Datenbankserver einen Fehler. Dieser Fehler wird als Fehlertext zurückgegeben.
	DbConnect()
-10	Ungültiger Wert für den Parameter Servertyp Die Funktion DbConnect() wurde mit einem nicht vereinbarten Servertyp aufgerufen. Gültige Werte sind 1 bis 4.
-11	Für das ausgewählte Datenbanksystem ist kein Provider installiert. Erforderlich: Zu dem angegeben Servertyp kann keine Verbindung aufgebaut werden, da auf dem PC nicht der erforderliche ADO.NET Provider installiert ist. In der Fehlermeldung wird der erforderliche Provider angegeben.
-12	Es kann keine Verbindung zum DBMS aufgebaut werden. Grund: Es konnte kein Verbindungsobjekt zu dem Datenbanksystem erzeugt werden. Die Ursache kann in einer fehlerhaften Verbindungszeichenkette liegen. Im Fehlertext wird der Grund für das Scheitern angegeben.
-13	Es kann keine Verbindung zum DBMS aufgebaut werden. Grund: Die Microsoft SQL Server Compact Edition Datenbank ist schreibgeschützt. Diese Fehlermeldung tritt nur bei der Microsoft SQL Server Compact Edition auf. Durch das Entfernen des Schreibschutzes der Datenbankdatei kann dieser Fehler behoben werden.
-14	Eine Verbindung zu der Serverversion wird nicht unterstützt. Die Zugriffe auf diese Serverversion werden nicht unterstützt, z.B. Oracle 8 oder Microsoft SQL Server 2000.
	DbSelect()
-20	Für folgende Spalten ist keine Konvertierungsregel implementiert ...

Fehlernummer	Beschreibung
	Es werden die Spalten aufgezählt, deren Datentyp nicht in FAMOS-Objekte konvertiert werden konnte. Zu diesen Datentypen zählen SqlGeography, SqlGeometry, SqlHierarchyId, SQLVariant vom Microsoft SQL Server. Für alle anderen Spalten wurden Datensätze bzw. Textfelder erzeugt.
-21	Da die Abfrage mehr als eine Ergebniszeile ergab, konnten die Blob-Spalten "Spaltennamen" nicht konvertiert werden. Wenn eine Abfrage gestellt wird, die Blob- Spalten enthält und mehrere Ergebniszeilen ergibt, dann können die Blob- Spalten nicht in FAMOS- Objekte konvertiert werden. Alle anderen Spalten der Abfrage sind als FAMOS- Objekte in der Ergebnisgruppe enthalten. Um eine Blob-Spalte zu lesen, muss die Abfrage so gestellt werden, dass nur eine Ergebniszeile entsteht.
-22	Die Zeitspalte xy existiert nicht im Abfrageergebnis. Eine Konvertierung in XY-Datensätze ist nicht möglich. Wird die Funktion DbSelect() mit dem Parameter TxTimeColumn aufgerufen und diese Spalte ist nicht im Abfrageergebnis vorhanden, so erfolgt keine Umwandlung in XY-Datensätze. Alle numerischen Spalten werden als normale Datensätze zurückgegeben.
-23	Die Zeitspalte xy ist kein normaler Datensatz. Die Angabe der Zeitspalte muss einer numerischen Spalte entsprechen, die in einen normalen Datensatz konvertiert werden kann. Ist das nicht der Fall, so erfolgt keine Umwandlung in XY-Datensätze. Alle numerischen Spalten werden als normale Datensätze zurückgegeben.
-24	Es gibt keine weiteren Datensätze, die in einen XY-Datensatz mit der Zeitspalte als X-Komponente umgewandelt werden können. Aus der Zeitspalte gibt es keine weiteren numerischen Spalten. Damit kann eine Umwandlung in XY- Datensätze nicht erfolgen.
	DbInsert(), DbUpdate1(), DbUpdate()
-30	Folgende Spalten existieren nicht in der Tabelle "Tabelle": "Spalten" In der Funktion DbInsert() wird eine Gruppe mit Gruppenelementen (Datensätze oder Textfelder) übergeben. Die Namen der Gruppenelemente werden gegen die Spaltennamen der Datenbanktabelle geprüft. Alle Gruppenelementnamen, die nicht einem Spaltennamen entsprechen, werden in der Fehlermeldung aufgezählt. Das Einfügen wird nicht ausgeführt.
-31	Die Gruppe enthält keine Elemente. Den Funktionen DbInsert(), DbUpdate1() oder DbUpdate() wurde eine leere FAMOS-Gruppe übergeben.
-32	Der Datentyp des Datensatzes xyz wird nicht unterstützt. Es ist nur ein "Normaler" Datensatz erlaubt In der übergebenen FAMOS- Gruppe ist ein Kanal, der nicht den Datentyp "Normaler Datensatz" hat. Es werden nur Kanäle mit dem Datentyp "Normaler Datensatz" unterstützt.

Fehlernummer	Beschreibung
-33	Konvertierungsfehler in Spalte "Spaltenname" : "Fehlertext" Vor dem Einfügen bzw. Aktualisieren der Daten in der Datenbank, werden die Inhalte der Gruppenelemente in den Datentyp der Spalte der Datenbanktabelle konvertiert. Tritt beim Konvertieren ein Fehler auf, so erscheint diese Fehlermeldung mit Angabe der Spalte und der Ursache.
-34	Die Gruppenelemente haben eine unterschiedliche Länge. Beim Einfügen bzw. Aktualisieren von Daten der Datenbank müssen alle Elemente der FAMOS- Gruppe die gleiche Größe haben.
-35	Der Datentyp "Datentyp" der Spalte "Spaltenname" wird nicht unterstützt. Es sollen Werte in eine Spalte der Datenbank eingefügt bzw. aktualisiert werden, dessen Datentyp nicht unterstützt wird. Zu diesen Datentypen zählen SqlGeography, SqlGeometry, SqlHierarchyId, SQLVariant vom Microsoft SQL Server. Das Einfügen bzw. Aktualisieren wird nicht ausgeführt.
-36	Fehler in der Update-Anweisung, Schlüsselwort "Schlüsselwort" nicht gefunden. In der SQL- Anweisung für die Funktionen DbUpdate1() bzw. DbUpdate() wurden die Schlüsselwörter UPDATE oder SET nicht gefunden.
-37	Fehler in der Update-Anweisung, keinen Tabellennamen gefunden In der SQL- Anweisung für die Funktionen DbUpdate1() bzw. DbUpdate() wurde kein Tabellename gefunden.
-38	Fehler in der Update-Anweisung, der SET- Teil ist leer In der SQL- Anweisung für die Funktionen DbUpdate1() bzw. DbUpdate() ist der SET- Teil leer.
-39	Fehler in der Update-Anweisung, Spalte "Spaltenname" existiert nicht. In der SQL- Anweisung für die Funktionen DbUpdate1() bzw. DbUpdate() ist im SET- Teil eine Spalte angegeben, die nicht existiert.
-40	Fehler in der Update-Anweisung in der Nähe von ... In der SQL- Anweisung für die Funktionen DbUpdate1() bzw. DbUpdate() ist in der WHERE- Bedingung ein Fehler aufgetreten. Der fehlerhafte Teil der WHERE- Bedingung wird in der Fehlermeldung angegeben.
-41	Für die Spalte "Spaltenname" existiert kein Gruppenelement In der SQL- Anweisung für die Funktionen DbUpdate1() bzw. DbUpdate() ist eine Spalte angegeben, für die kein gleichnamiges Gruppenelement existiert.
-42	Der Sample-Index ist 0 oder größer als die Länge der Gruppenelemente Der Parameter Sample Index in der Funktion DbUpdate1() liegt außerhalb des Bereichs (1... Maximale Anzahl von Werten eines Gruppenelements)
-43	Das in der Update-Anweisung enthaltene Gruppenelement ... ist nicht in der Gruppe vorhanden.
	DbOption()

Fehlernummer	Beschreibung
-50	Der optionale Parameter "Parametername" ist nicht vereinbart. In der Funktion DbOption() wurde ein nicht vereinbarter Parametername angegeben.
-51	Der Timeout- Wert ist ungültig. Angabe 1... in Sekunden Der optionale Parameter „CommandTimeOut“ wurde mit einem ungültigen Wert gesetzt. Die Angabe muss ganzzahlig sein und erfolgt in Sekunden.
	Interne Fehler
-300	Interner Fehler: DBConnection- Objekt ist nicht vorhanden Beim Öffnen einer Verbindung oder beim Erzeugen eines Datenbankkommandos ist kein Verbindungsobjekt vorhanden.
-301	Interner Fehler: Keine Datenquelle vorhanden Beim Ausführen einer SQL- Anweisung ist keine Datenquelle vorhanden.
	Ausnahmefehler in den Kit- Funktionen
-400	Ausnahme in DbConnect: "Grund" In der Funktion ist eine unerwartete Ausnahme aufgetreten.
-401	Ausnahme in DbDisconnect: "Grund" In der Funktion ist eine unerwartete Ausnahme aufgetreten.
-402	Ausnahme in DbInitialize: "Grund" In der Funktion ist eine unerwartete Ausnahme aufgetreten.
-500	Ausnahme in DbSql: "Grund" In der Funktion ist eine unerwartete Ausnahme aufgetreten.
-501	Ausnahme in DbSelect: "Grund" In der Funktion ist eine unerwartete Ausnahme aufgetreten.
-502	Ausnahme in DbInsert: "Grund" In der Funktion ist eine unerwartete Ausnahme aufgetreten.
-503	Ausnahme in DbUpdate1: "Grund" In der Funktion ist eine unerwartete Ausnahme aufgetreten.
-504	Ausnahme in DbUpdate: "Grund" In der Funktion ist eine unerwartete Ausnahme aufgetreten.
-600	Ausnahme in DbBeginTransaction: "Grund" In der Funktion ist eine unerwartete Ausnahme aufgetreten.
-601	Ausnahme in DbEndTransaction: "Grund" In der Funktion ist eine unerwartete Ausnahme aufgetreten.
-700	Ausnahme in DbGetLastErrorText: "Grund" In der Funktion ist eine unerwartete Ausnahme aufgetreten.

Fehlernummer	Beschreibung
-701	Ausnahme in DbGetLastErrorCode: "Grund" In der Funktion ist eine unerwartete Ausnahme aufgetreten.
-702	Ausnahme in DbOption: "Grund" In der Funktion ist eine unerwartete Ausnahme aufgetreten.

Index

A

AGB 6
Allgemeinen Geschäftsbedingungen 6
Änderungswünsche 6

B

Behandlung von Nullwerten 36

C

CE-Konformität 6

D

Datenbankverbindung 12
DbBeginTransaction 28
DbConnect 13
DbDisconnect 16
DbEndTransaction 30
DbGetLastErrorCode 32
DbGetLastErrorText 30
DbInitialize 13
DbInsert 20
DbOption 33
DbSelect 18
DbSql 17
DbUpdate 26
DbUpdate1 23
DIN-EN-ISO-9001 6

F

Fehlercodes 37
Fehlermeldungen 6

G

Gewährleistung 6

H

Haftungsbeschränkung 6
Hotline
Technischer Support 5

I

imc Software-Lizenzvertrag 7
ISO-9001 6

J

Justage 6

K

Kalibrierung 6

Konvertierung der Daten 34
Kundendienst
Technischer Support 5

Q

Qualitätsmanagement 6

R

Reparatur 6

S

Service
Technischer Support 5
Service und Wartung 6
Service-Check 6

T

Technischer Support 5
Telefonnummer
Technischer Support 5
Transaktionen 28

U

Unterstützte Datenbanken 10

W

Wartung 6

Z

Zertifikate 6
Zugriff auf die Datenbank 17



An Axiometrix Solutions Brand

Kontaktaufnahme mit imc

Adresse

imc Test & Measurement GmbH
Voltastraße 5
13355 Berlin

Telefon: +49 30 467090-0
E-Mail: info@imc-tm.de
Internet: <https://www.imc-tm.de>

Technischer Support

Zur technischen Unterstützung steht Ihnen unser technischer Support zur Verfügung:

Telefon: +49 30 467090-26
E-Mail: hotline@imc-tm.de
Internet: <https://www.imc-tm.de/service-training/>

Service und Wartung

Für Service- und Wartungsanfragen steht Ihnen unser Serviceteam zur Verfügung:

Telefon: +49 30 629396-333
E-Mail: imc-service@axiomatrixsolutions.com
Internet: <https://www.imc-tm.de/service>

imc ACADEMY - Trainingscenter

Der sichere Umgang mit Messgeräten erfordert gute Systemkenntnisse. In unserem Trainingscenter werden diese von erfahrenen Messtechnik Spezialisten vermittelt.

E-Mail: schulung@imc-tm.de
Internet: <https://www.imc-tm.de/service-training/imc-academy>

Internationale Vertriebspartner

Den für Sie zuständigen Ansprechpartner, finden Sie in unserer Übersichtsliste der imc Partner:

Internet: <https://www.imc-tm.de/imc-weltweit/>

imc @ Social Media

<https://www.facebook.com/imcTestMeasurement>

<https://www.youtube.com/c/imcTestMeasurementGmbH>

https://x.com/imc_de

<https://www.linkedin.com/company/imc-test-&-measurement-gmbh>