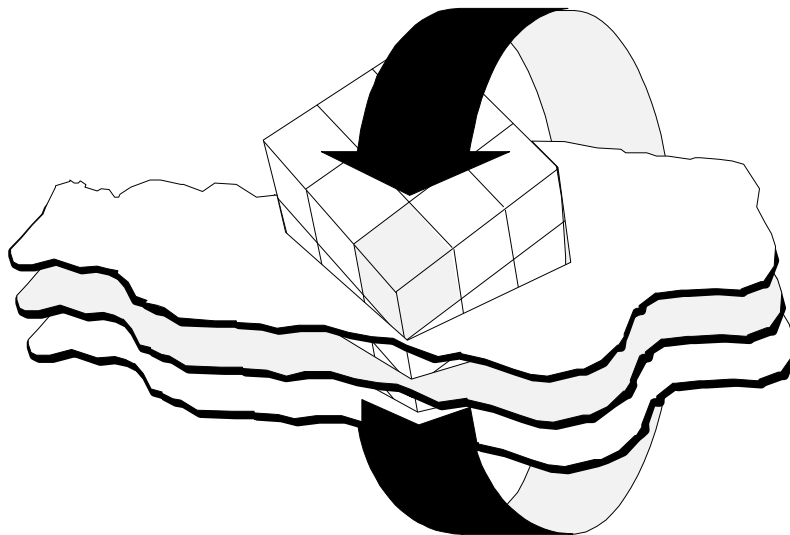


OLE- Automation

Komponentenstruktur für den Analyse-Assistent



Version 0.4

Inhaltsverzeichnis

1 ARCHITEKTUR	5
<hr/>	
1.1 ALLGEMEIN	5
1.2 OCX-KOMPONENTE	6
1.2.1 HORIZONTALE UNTERTEILUNG	6
1.2.2 VERTIKALE UNTERTEILUNG	7
2 OBJEKTE UND IHRE METHODEN	8
<hr/>	
2.1 METHODENNAME	8
2.2 METHODENAUFTRUF	8
3 METHODEN FÜR MENGEN-CONTAINER (SETCONTAINER)	9
<hr/>	
3.1 SETCURRENTUSER()	9
3.2 GETCURRENTUSER()	9
3.3 SETTYPEFILTER()	10
3.4 SETENTITYFILTER()	10
3.5 SETPROJECTFILTER()	10
3.6 SETNAMEFILTER()	11
3.7 SETOWNERFILTER()	11
3.8 SETDATEFILTER()	11
3.9 RESETALLFILTER()	12
3.10 RESETTYPEFILTER()	12
3.11 RESETENTITYFILTER()	12
3.12 RESETPROJECTFILTER()	13
3.13 RESETNAMEFILTER()	13
3.14 RESETOWNERFILTER()	13
3.15 RESETDATEFILTER()	14
3.16 CREATESETLIST()	14
3.17 FREESSETLIST()	15
3.18 GETCOUNT()	15
3.19 GETFIRST()	16
3.20 GETNEXT()	16

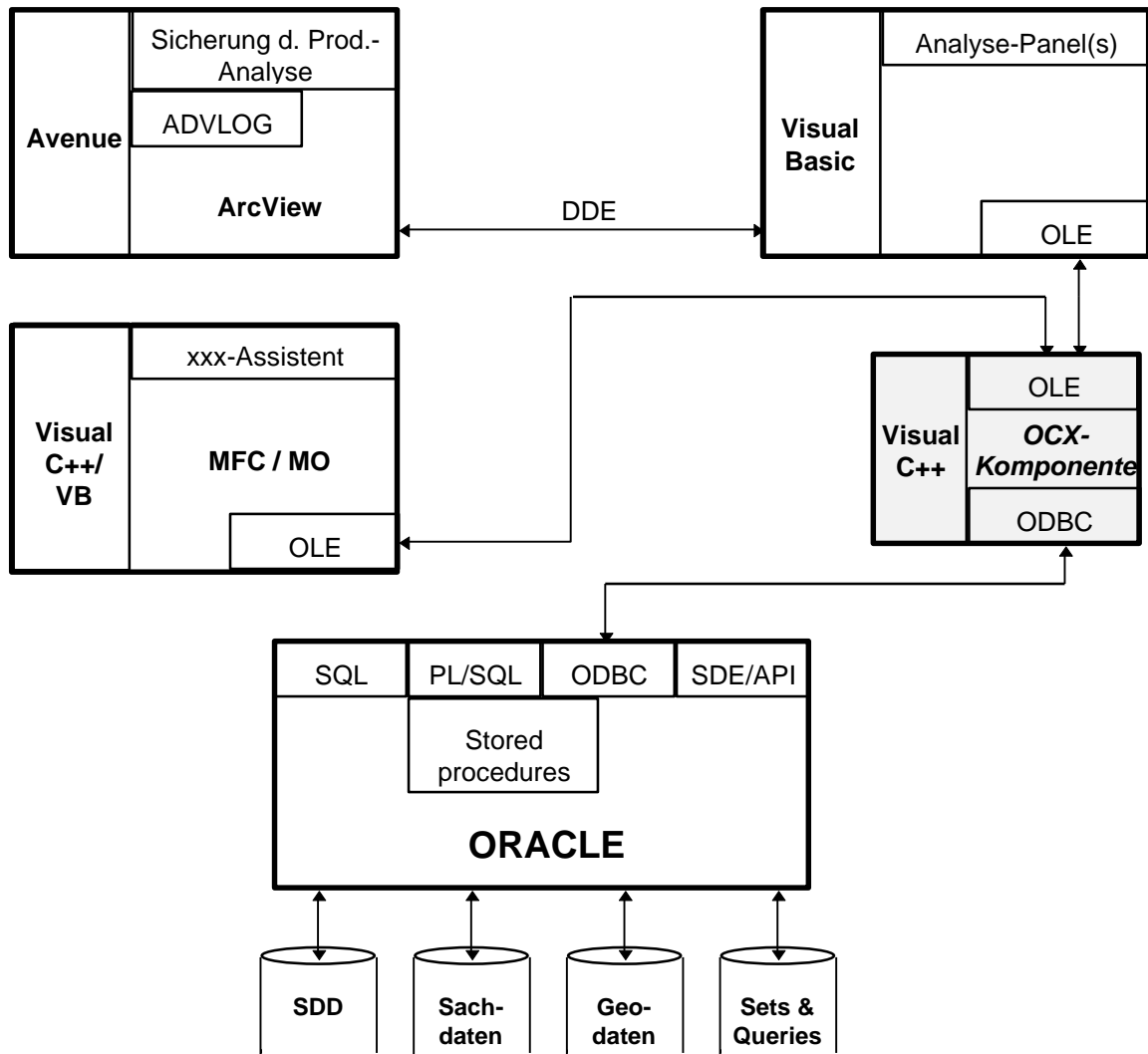
4 METHODEN FÜR MENGEN-OBJEKT (SETSET)	17
4.1 INIT()	17
4.2 GETID()	17
4.3 GETSCOPE()	18
4.4 GETTYPE ()	18
4.5 GETENTITY()	19
4.6 GETPROJECT()	19
4.7 GETCOUNT()	20
4.8 GETSTATUS()	20
4.9 GETNOOFATTRIBUTES()	21
4.10 GETNAME()	21
4.11 GETOWNER()	22
4.12 GETDATE()	22
4.13 GETDESCRIPTION()	23
4.14 CHECKACCESS()	23
4.15 CHECKSUBSET()	24
4.16 INITNEW()	24
4.17 CHANGETYPETORELATIONSET()	25
4.18 CHANGETYPETOATTRIBUTESET()	25
4.19 INSERTELEMENTS()	26
4.20 INSERTSINGLEELEMENT()	26
4.21 INSERTMULTIPLEELEMENTS()	27
4.22 DELETEELEMENT()	27
4.23 DELETEMULTIPLEELEMENTS()	28
4.24 COPY ()	28
4.25 TRANSFORM ()	29
4.26 MERGEAND()	29
4.27 MERGEOR()	30
4.28 MERGEMINUS()	31
4.29 COMMIT()	32
4.30 ROLLBACK ()	33
4.31 LOCK ()	33
4.32 SETSCOPE()	34
4.33 SETENTITY()	34
4.34 SETPROJECT()	35
4.35 SETNAME()	35
4.36 SETDESCRIPTION()	36
4.37 SETNAMEREF()	36
4.38 DELETE ()	37
4.39 RESET()	37
4.40 GETFIRSTELEMENT()	38
4.41 GETNEXTELEMENT()	38
4.42 CREATEATTRIBUTEList()	39
4.43 FREEATTRIBUTEList()	39
4.44 GETFIRSTATTRIBUTE()	40
4.45 GETNEXTATTRIBUTE()	40

5 METHODEN FÜR ATTRIBUT-OBJEKT (SETATTRIBUTE)	41
5.1 INIT()	41
5.2 INITNEWRBS()	41
5.3 INITNEWARITHMETIC()	42
5.4 INITNEWAVERAGE()	43
5.5 INITNEWSUM()	43
5.6 INITNEWMAXIMUM()	44
5.7 INITNEWMINIMUM()	44
5.8 INITNEWCOUNT()	45
5.9 INITNEWSTDDEV()	45
5.10 INITNEWVARIANCE()	46
5.11 SETNAME()	46
5.12 SETDESCRIPTION()	47
5.13 GETTYPE()	47
5.14 GETLENGTH()	48
5.15 GETPERCISION()	48
5.16 GETNAME()	49
5.17 GETDESCRIPTION()	49
5.18 GETMINMAXFLOAT()	50
5.19 GETMINMAXINTEGER()	50
5.20 GETMINMAXSTRING()	51
5.21 GETMULTIPLEMINMAXFLOAT()	51
6 RETURN CODES	52
7 OPERATIONEN	53
7.1 OPERATIONEN MIT MENGEN	53
7.1.1 MENGE ERZEUGEN	53
7.1.2 MENGE KOPIEREN	54
7.1.3 MENGE LÖSCHEN	54
7.1.4 ATTRIBUTE ERZEUGEN	55

1 Architektur

1.1 Allgemein

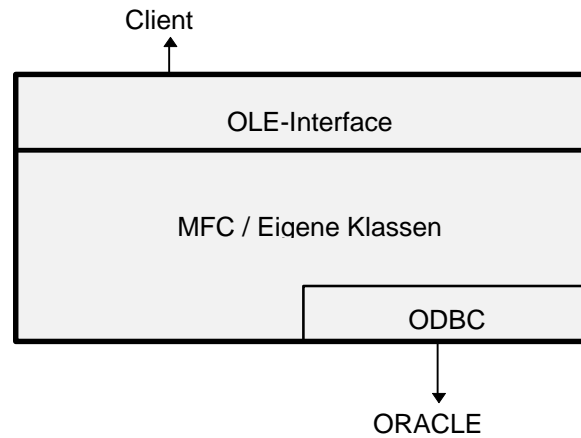
Die Kommunikation zwischen den GIS-Analyse-Assistenten und der Datenbank erfolgt über ODBC. Dazwischen wird eine OCX-Komponente eingefügt, welche Anfragen vom Assistenten über eine vordefinierte (OLE-) Schnittstelle abarbeitet (z.B. gib mir alle Mengen oder Entitäten die im SDD gespeichert sind).



1.2 OCX-Komponente

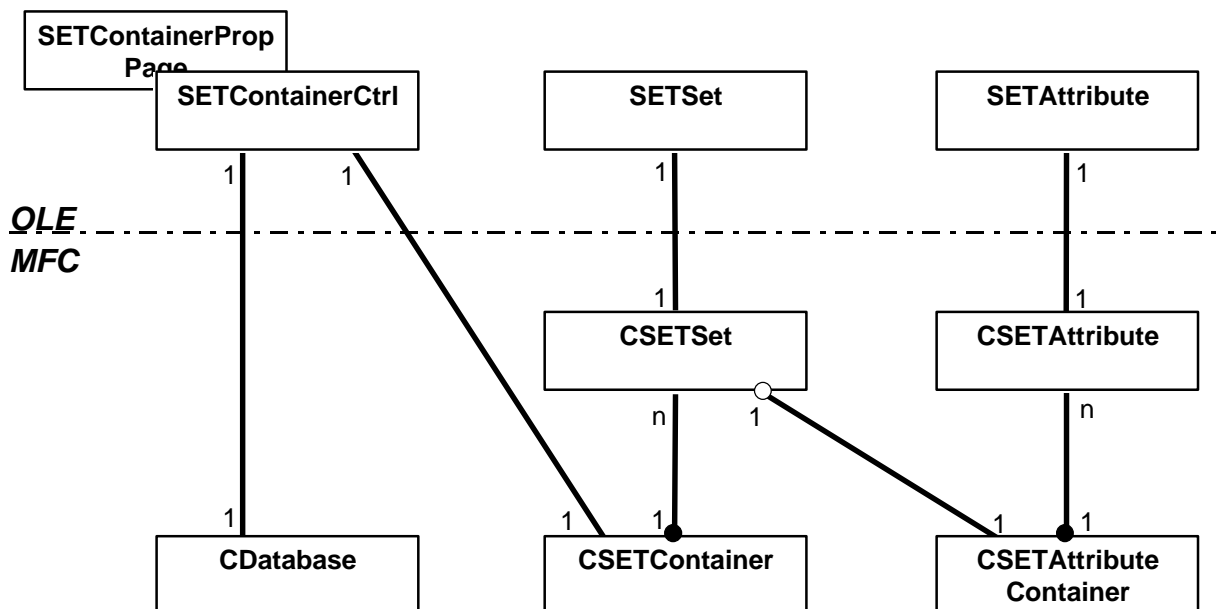
Die OCX-Komponente ist horizontal in verschiedene Schichten und vertikal in unterschiedliche Anwendungsblöcke unterteilt.

1.2.1 Horizontale Unterteilung



1.2.1.1 Mengen

Das Klassenmodell sieht wie folgt aus:



1.2.2 Vertikale Unterteilung

Semantic Data Dictionary (SDD)	Mengen (SET)	Abfragen (QRY)	...
--	------------------------	--------------------------	-----

Der Anwendungsblock SDD ist der Kern der OCX-Komponente und wird von allen anderen Blöcken verwendet. D.h die Methoden von der Klasse des Semantic Data Dictionary (SDD) werden von Mengen und Abfragen aufgerufen.

2 Objekte und ihre Methoden

2.1 Methodenname

Der Methodenname ist gegliedert in:

- *Aktion* (Get, Set, Delete, Create, ...)
- *Subjekt* (element, filter, integer, ...)

2.2 Methodenaufruf

Nachfolgend werden nur die Methoden aufgelistet. Der Aufruf hat mit der vollen Syntax zu erfolgen (z.B. SETContainer.SetCurrentUser(...)).

3 Methoden für Mengen-Container (SETContainer)

3.1 SetCurrentUser()

Interface	<code>long SetCurrentUser(BSTR FAR* user, int dba_flg)</code>						
Parameters	<code>user</code> - Ein Zeiger zum aktuellen (Login-) user in der Anwendung. <code>dba_flg</code> - Ein Flag für DBA-Rechte (1) oder normale Benutzerrechte (0).						
Description	Diese Methode muss vor jeder anderen SETContainer - Methode aufgerufen werden. Sie teilt dem Anwendungsblock SET den aktuellen User und das entsprechende DBA-Flag mit. Diese Informationen werden für Prüfungen von Rechten und bei der Anlegung einer neuen Menge verwendet. Ist das DBA-Flag gesetzt, so darf der aktuelle User alle Mengen löschen, ändern, etc.						
Returns	<table> <tr> <td>SET_FAILURE</td> <td>(-14)</td> <td>Bei einem Fehler.</td> </tr> <tr> <td>SET_NO_ERROR</td> <td>(0)</td> <td>Bei keinem Fehler.</td> </tr> </table>	SET_FAILURE	(-14)	Bei einem Fehler.	SET_NO_ERROR	(0)	Bei keinem Fehler.
SET_FAILURE	(-14)	Bei einem Fehler.					
SET_NO_ERROR	(0)	Bei keinem Fehler.					

3.2 GetCurrentUser()

Interface	<code>long GetCurrentUser(BSTR FAR* currUser)</code>						
Parameters	<code>currUser</code> - Platzhalter, in den der Wert geschrieben wird.						
Description	Diese Methode liefert den aktuellen User zurück.						
Returns	<table> <tr> <td>SET_FAILURE</td> <td>(-14)</td> <td>Bei einem Fehler.</td> </tr> <tr> <td>SET_NO_ERROR</td> <td>(0)</td> <td>Bei keinem Fehler.</td> </tr> </table>	SET_FAILURE	(-14)	Bei einem Fehler.	SET_NO_ERROR	(0)	Bei keinem Fehler.
SET_FAILURE	(-14)	Bei einem Fehler.					
SET_NO_ERROR	(0)	Bei keinem Fehler.					

3.3 SetTypeFilter()

Interface	<code>long SetTypeFilter(integer value)</code>		
Parameters	<code>value</code> - Wert mit dem der Filter geSetzt werden soll.		
Description	Mit dieser Methode kann die Filteroption für den Mengentyp (Objektmenge =1, Attributmenge =2, Beziehungsmenge =3, Infopaket =4) geSetzt werden. D.h. die Funktion <code>CreateSetList()</code> berücksichtigt diesen Filter.		
Returns	<code>SET_FAILURE</code>	(-14)	Bei einem Fehler.
	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.

3.4 SetEntityFilter()

Interface	<code>long SetEntityFilter(integer value)</code>		
Parameters	<code>value</code> - Wert mit dem der Filter geSetzt werden soll.		
Description	Mit dieser Methode kann die Filteroption für die Entitätszugehörigkeit der Menge (ID der Entität) geSetzt werden. D.h. die Funktion <code>CreateSetList()</code> berücksichtigt diesen Filter.		
Returns	<code>SET_FAILURE</code>	(-14)	Bei einem Fehler.
	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.

3.5 SetProjectFilter()

Interface	<code>long SetProjectFilter(integer value)</code>		
Parameters	<code>value</code> - Wert mit dem der Filter geSetzt werden soll.		
Description	Mit dieser Methode kann die Filteroption für die Projektzugehörigkeit von Mengen (ID der langen Transaktion) geSetzt werden. D.h. die Funktion <code>CreateSetList()</code> berücksichtigt diesen Filter.		
Returns	<code>SET_FAILURE</code>	(-14)	Bei einem Fehler.
	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.

3.6 SetNameFilter()

Interface	<code>long SetNameFilter(BSTR FAR* value)</code>		
Parameters	<code>value</code> - Wert mit dem der Filter geSetzt werden soll.		
Description	Mit dieser Methode kann die Filteroption für den Mengename (mit Wildcard) geSetzt werden. D.h. die Funktion <code>CreateSetList()</code> berücksichtigt diesen Filter.		
Returns	<code>SET_FAILURE</code>	(-14)	Bei einem Fehler.
	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.

3.7 SetOwnerFilter()

Interface	<code>long SetOwnerFilter(BSTR FAR* value)</code>		
Parameters	<code>value</code> - Wert mit dem der Filter geSetzt werden soll.		
Description	Mit dieser Methode kann die Filteroption für den Mengenbesitzer geSetzt werden. D.h. die Funktion <code>CreateSetList()</code> berücksichtigt diesen Filter.		
Returns	<code>SET_FAILURE</code>	(-14)	Bei einem Fehler.
	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.

3.8 SetDateFilter()

Interface	<code>long SetDateFilter(BSTR FAR* value)</code>		
Parameters	<code>value</code> - Wert mit dem der Filter geSetzt werden soll.		
Description	Mit dieser Methode kann die Filteroption für das Erstellungsdatum der Menge geSetzt werden. D.h. die Funktion <code>CreateSetList()</code> berücksichtigt diesen Filter.		
Returns	<code>SET_FAILURE</code>	(-14)	Bei einem Fehler.
	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.

3.9 ResetAllFilter()

Interface	<code>long ResetAllFilter()</code>		
Parameters	-		
Description	Mit dieser Methode werden alle Filteroptionen zurückgeSetzt. D.h. die Funktion <code>CreateSetList()</code> berücksichtigt diesen Filter.		
Returns	<code>SET_FAILURE</code>	(-14)	Bei einem Fehler.
	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.

3.10 ResetTypeFilter()

Interface	<code>long ResetTypeFilter()</code>		
Parameters	-		
Description	Mit dieser Methode wird die Filteroptionen für den Mengentyp zurückgeSetzt. D.h. die Funktion <code>CreateSetList()</code> berücksichtigt diesen Filter.		
Returns	<code>SET_FAILURE</code>	(-14)	Bei einem Fehler.
	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.

3.11 ResetEntityFilter()

Interface	<code>long ResetEntityFilter()</code>		
Parameters	-		
Description	Mit dieser Methode wird die Filteroptionen für die Entitätszugehörigkeit der Menge zurückgeSetzt. D.h. die Funktion <code>CreateSetList()</code> berücksichtigt diesen Filter.		
Returns	<code>SET_FAILURE</code>	(-14)	Bei einem Fehler.
	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.

3.12 ResetProjectFilter()

Interface	<code>long ResetProjectFilter()</code>		
Parameters	-		
Description	Mit dieser Methode wird die Filteroptionen für die Projektgehörigkeit der Menge (ID der langen Transaktion) zurückgeSetzt. D.h. die Funktion <code>CreateSetList()</code> berücksichtigt diesen Filter.		
Returns	<code>SET_FAILURE</code>	(-14)	Bei einem Fehler.
	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.

3.13 ResetNameFilter()

Interface	<code>long ResetNameFilter()</code>		
Parameters	-		
Description	Mit dieser Methode wird die Filteroptionen für den Mengennamen zurückgeSetzt. D.h. die Funktion <code>CreateSetList()</code> berücksichtigt diesen Filter.		
Returns	<code>SET_FAILURE</code>	(-14)	Bei einem Fehler.
	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.

3.14 ResetOwnerFilter()

Interface	<code>long ResetOwnerFilter()</code>		
Parameters	-		
Description	Mit dieser Methode wird die Filteroptionen für den Besitzer der Menge zurückgeSetzt. D.h. die Funktion <code>CreateSetList()</code> berücksichtigt diesen Filter.		
Returns	<code>SET_FAILURE</code>	(-14)	Bei einem Fehler.
	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.

3.15 ResetDateFilter()

Interface	long ResetDateFilter()		
Parameters	-		
Description	Mit dieser Methode wird die Filteroptionen für das Erstellungsdatum der Menge zurückgeSetzt werden. D.h. die Funktion CreateSetList() berücksichtigt diesen Filter.		
Returns	SET_FAILURE	(-14)	Bei einem Fehler.
	SET_NO_ERROR	(0)	Bei keinem Fehler.

3.16 CreateSetList()

Interface	long CreateSetList()		
Parameters	-		
Description	Mit dieser Methode wird über ODBC ein Connect zu der Datenbank hergestellt. Gemäss den geSetzten Filter (Set...Filter()) werden aus der Datenbank die gewünschten Mengen aus der Verwaltungstabelle gelesen und im Memory abgelegt.		
Returns	SET_NO_DB_CONNECT	(-9)	Keine Verbindung zur Datenbank.
	DB_SQL_ERROR	(-2)	Internes SQL-Statement ist falsch.
	SET_NO_ERROR	(0)	Bei keinem Fehler.
Notes	Mit den Methoden GetFirst() und GetNext() kann der Name der Menge, in der aktuellen Liste, gelesen werden.		

3.17 FreeSetList()

Interface	<code>void FreeSetList()</code>
Parameters	-
Description	Diese Methode gibt den Speicher frei, der in der Funktion <code>CreateSetList()</code> alloziert worden ist.
Returns	-
Notes	Vorgängig muss die Funktion <code>CreateSetList()</code> aufgerufen worden sein.

3.18 GetCount()

Interface	<code>long GetCount(long FAR* num)</code>									
Parameters	<code>num</code> - Platzhalter, für die Anzahl der Listeneinträge									
Description	Diese Funktion liefert die Anzahl der Mengen zurück, die gemäss der Filter-Restirktion selektiert worden sind.									
Returns	<table> <tr> <td><code>SET_LIST_NOT_INIT</code></td> <td>(-1)</td> <td>Die Liste ist vorgängig nicht erzeugt worden.</td> </tr> <tr> <td><code>SET_LIST_IS_EMPTY</code></td> <td>(-2)</td> <td>Die Liste ist leer.</td> </tr> <tr> <td><code>SET_NO_ERROR</code></td> <td>(0)</td> <td>Bei keinem Fehler.</td> </tr> </table>	<code>SET_LIST_NOT_INIT</code>	(-1)	Die Liste ist vorgängig nicht erzeugt worden.	<code>SET_LIST_IS_EMPTY</code>	(-2)	Die Liste ist leer.	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.
<code>SET_LIST_NOT_INIT</code>	(-1)	Die Liste ist vorgängig nicht erzeugt worden.								
<code>SET_LIST_IS_EMPTY</code>	(-2)	Die Liste ist leer.								
<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.								
Notes	Die Information der Anzahl von Mengen kann mittels einem Loop mit der Funktion <code>GetFirst()</code> und <code>GetNext()</code> genutzt werden.									

3.19 GetFirst()

Interface	<code>long GetFirst(BSTR FAR* name)</code>		
Parameters	<code>name</code> - Platzhalter, in welchen der erste Mengename in der Liste geschrieben wird.		
Description	Mit dieser Methode wird der erste Mengename in der Liste zurückgegeben.		
Returns	<code>SET_LIST_NOT_INIT</code>	(-1)	Die Liste ist vorgängig nicht erzeugt worden.
	<code>SET_OBJ_NOT_FOUND</code>	(-3)	Kein Objekt gefunden.
	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.
Notes	Die Methode <code>CreateSetList()</code> muss vorgängig aufgerufen worden sein.		

3.20 GetNext()

Interface	<code>long GetNext(BSTR FAR* name)</code>		
Parameters	<code>name</code> - Platzhalter, in welchen der Mengename in der Liste geschrieben wird.		
Description	Mit dieser Methode wird jeweils der nächste Mengename in der Liste zurückgegeben.		
Returns	<code>SET_LIST_NOT_INIT</code>	(-1)	Die Liste ist vorgängig nicht erzeugt worden.
	<code>SET_EOF</code>	(1)	Listenende erreicht..
	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.
Notes	Die Methoden <code>CreateSetList()</code> und <code>GetFirst()</code> müssen vorgängig aufgerufen worden sein.		

4 Methoden für Mengen-Objekt (SETSet)

4.1 Init()

Interface	<code>long Init(long pos)</code>		
Parameters	<code>pos</code> - Positionsindex der Menge in der Liste		
Description	Diese Methode Initialisiert, gemäss dem Positionsindex in der Liste, ein Objekt der Klasse SETSet.		
Returns	<code>SET_LIST_NOT_INIT</code>	(-1)	Die Liste ist vorgängig nicht erzeugt worden.
	<code>SET_LIST_IS_EMPTY</code>	(-2)	Die Liste ist leer.
	<code>SET_OBJ_NOT_FOUND</code>	(-3)	Kein Objekt gefunden.
	<code>SET_POS_IS_WRONG</code>	(-4)	Angegebene Position ist falsch.
	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.
Notes	Die Liste muss vorher mit <code>CreateSetList()</code> erzeugt worden sein.		

4.2 GetId()

Interface	<code>long GetId(long FAR* id)</code>		
Parameters	<code>id</code> - Platzhalter, in welchen die ID der Menge geschrieben wird.		
Description	Diese Methode liefert die ID der Menge zurück.		
Returns	<code>SET_OBJ_NOT_FOUND</code>	(-3)	Bei einem Fehler.
	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.
Notes	Das SETSet-Objekt muss vorher mit <code>Init()</code> Initialisiert worden sein.		

4.3 GetScope()

Interface	<code>long GetScope(long FAR* value)</code>						
Parameters	<code>value</code> - Platzhalter, in welchen der Besitzstand geschrieben wird.						
Description	Mit dieser Methode kann der Besitzstand der Menge abgefragt werden.						
Returns	<table> <tr> <td><code>SET_OBJ_NOT_FOUND</code></td> <td>(-3)</td> <td>Bei einem Fehler.</td> </tr> <tr> <td><code>SET_NO_ERROR</code></td> <td>(0)</td> <td>Bei keinem Fehler.</td> </tr> </table>	<code>SET_OBJ_NOT_FOUND</code>	(-3)	Bei einem Fehler.	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.
<code>SET_OBJ_NOT_FOUND</code>	(-3)	Bei einem Fehler.					
<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.					
Notes	Das SETSet-Objekt muss vorher mit <code>Init()</code> Initialisiert worden sein.						

4.4 GetType ()

Interface	<code>long GetType(long FAR* value)</code>						
Parameters	<code>value</code> - Platzhalter, in welchen der Typ geschrieben wird.						
Description	Mit dieser Methode kann der Type der Menge (Objektmenge, Attributmenge, Beziehungsmenge, Infopaket) abgefragt werden.						
Returns	<table> <tr> <td><code>SET_OBJ_NOT_FOUND</code></td> <td>(-3)</td> <td>Bei einem Fehler.</td> </tr> <tr> <td><code>SET_NO_ERROR</code></td> <td>(0)</td> <td>Bei keinem Fehler.</td> </tr> </table>	<code>SET_OBJ_NOT_FOUND</code>	(-3)	Bei einem Fehler.	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.
<code>SET_OBJ_NOT_FOUND</code>	(-3)	Bei einem Fehler.					
<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.					
Notes	Das SETSet-Objekt muss vorher mit <code>Init()</code> Initialisiert worden sein.						

4.5 GetEntity()

Interface	<code>long GetEntity(long FAR* value)</code>						
Parameters	<code>value</code> - Platzhalter, in welchen die Entitäts-ID geschrieben wird.						
Description	Mit dieser Methode kann Entitäts-ID der Menge abgefragt werden.						
Returns	<table> <tr> <td><code>SET_OBJ_NOT_FOUND</code></td> <td>(-3)</td> <td>Bei einem Fehler.</td> </tr> <tr> <td><code>SET_NO_ERROR</code></td> <td>(0)</td> <td>Bei keinem Fehler.</td> </tr> </table>	<code>SET_OBJ_NOT_FOUND</code>	(-3)	Bei einem Fehler.	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.
<code>SET_OBJ_NOT_FOUND</code>	(-3)	Bei einem Fehler.					
<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.					
Notes	Das SETSet-Objekt muss vorher mit <code>Init()</code> Initialisiert worden sein.						

4.6 GetProject()

Interface	<code>long GetProject(long FAR* value)</code>						
Parameters	<code>value</code> - Platzhalter, in welchen die Projekt-ID geschrieben wird.						
Description	Mit dieser Methode kann Projekt-ID der Menge (ID der langen Transaktion) abgefragt werden.						
Returns	<table> <tr> <td><code>SET_OBJ_NOT_FOUND</code></td> <td>(-3)</td> <td>Bei einem Fehler.</td> </tr> <tr> <td><code>SET_NO_ERROR</code></td> <td>(0)</td> <td>Bei keinem Fehler.</td> </tr> </table>	<code>SET_OBJ_NOT_FOUND</code>	(-3)	Bei einem Fehler.	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.
<code>SET_OBJ_NOT_FOUND</code>	(-3)	Bei einem Fehler.					
<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.					
Notes	Das SETSet-Objekt muss vorher mit <code>Init()</code> Initialisiert worden sein.						

4.7 GetCount()

Interface	<code>long GetCount(long FAR* value)</code>						
Parameters	<code>value</code> - Platzhalter, in welchen die Anzahl der Elemente der Menge geschrieben wird.						
Description	Mit dieser Methode kann die Anzahl der Elemente der Menge abgefragt werden.						
Returns	<table> <tr> <td><code>SET_OBJ_NOT_FOUND</code></td> <td>(-3)</td> <td>Bei einem Fehler.</td> </tr> <tr> <td><code>SET_NO_ERROR</code></td> <td>(0)</td> <td>Bei keinem Fehler.</td> </tr> </table>	<code>SET_OBJ_NOT_FOUND</code>	(-3)	Bei einem Fehler.	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.
<code>SET_OBJ_NOT_FOUND</code>	(-3)	Bei einem Fehler.					
<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.					
Notes	Das SETSet-Objekt muss vorher mit <code>Init()</code> Initialisiert worden sein.						

4.8 GetStatus()

Interface	<code>long GetStatus(long FAR* value)</code>						
Parameters	<code>value</code> - Platzhalter, in welchen der Status der Menge geschrieben wird.						
Description	Mit dieser Methode kann der Status der Menge (1: Initialisiert, 2: Referenziert durch Element-ID, 3: Referenziert durch den Elementnamen) abgefragt werden.						
Returns	<table> <tr> <td><code>SET_OBJ_NOT_FOUND</code></td> <td>(-3)</td> <td>Bei einem Fehler.</td> </tr> <tr> <td><code>SET_NO_ERROR</code></td> <td>(0)</td> <td>Bei keinem Fehler.</td> </tr> </table>	<code>SET_OBJ_NOT_FOUND</code>	(-3)	Bei einem Fehler.	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.
<code>SET_OBJ_NOT_FOUND</code>	(-3)	Bei einem Fehler.					
<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.					
Notes	Das SETSet-Objekt muss vorher mit <code>Init()</code> Initialisiert worden sein.						

4.9 GetNoOfAttributes()

Interface	<code>long GetNoOfAttributes(long FAR* num)</code>		
Parameters	<code>num</code> - Platzhalter, für die Anzahl der Attribute der Menge		
Description	Diese Funktion liefert die Anzahl der Attribute zurück.		
Returns	<code>SET_OBJ_NOT_FOUND</code>	(-3)	Bei einem Fehler.
	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.
Notes	Die Information der Anzahl von Mengen kann mittels einem Loop mit der Funktion <code>GetFirstAttribute()</code> und <code>GetNextAttribute()</code> genutzt werden.		

4.10 GetName()

Interface	<code>long GetName(BSTR FAR* value)</code>		
Parameters	<code>value</code> - Platzhalter, in welchen der Name geschrieben wird.		
Description	Mit dieser Methode kann der Name der Menge abgefragt werden.		
Returns	<code>SET_OBJ_NOT_FOUND</code>	(-3)	Bei einem Fehler.
	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.
Notes	Das SETSet-Objekt muss vorher mit <code>Init()</code> Initialisiert worden sein.		

4.11 GetOwner()

Interface	<code>long GetOwner(BSTR FAR* value)</code>		
Parameters	<code>value</code> - Platzhalter, in welchen der Besitzer geschrieben wird.		
Description	Mit dieser Methode kann der Besitzer der Menge abgefragt werden.		
Returns	<code>SET_OBJ_NOT_FOUND</code>	(-3)	Bei einem Fehler.
	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.
Notes	Das SETSet-Objekt muss vorher mit <code>Init()</code> Initialisiert worden sein.		

4.12 GetDate()

Interface	<code>long GetDate(BSTR FAR* value)</code>		
Parameters	<code>value</code> - Platzhalter, in welchen das Erstellungsdatum geschrieben wird.		
Description	Mit dieser Methode kann das Erstellungsdatum der Menge abgefragt werden.		
Returns	<code>SET_OBJ_NOT_FOUND</code>	(-3)	Bei einem Fehler.
	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.
Notes	Das SETSet-Objekt muss vorher mit <code>Init()</code> Initialisiert worden sein.		

4.13 GetDescription()

Interface	<code>long GetDescription(BSTR FAR* value)</code>		
Parameters	<code>value</code> - Platzhalter, in welchen die Beschreibung geschrieben wird.		
Description	Mit dieser Methode kann die Beschreibung der Menge abgefragt werden.		
Returns	<code>SET_OBJ_NOT_FOUND</code>	(-3)	Bei einem Fehler.
	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.
Notes	Das SETSet-Objekt muss vorher mit <code>Init()</code> Initialisiert worden sein.		

4.14 CheckAccess()

Interface	<code>long CheckAccess(long FAR* value)</code>		
Parameters	<code>value</code> - Platzhalter, in welchen die Berechtigung geschrieben wird. <ul style="list-style-type: none"> • <code>WRITE_ACCESS</code> (1) == Schreib-Berechtigung • <code>READ_ACCESS</code> (2) == Lese-Berechtigung • <code>NO_ACCESS</code> (3) == Keine Berechtigung 		
Description	Diese Funktion liefert die Rechte zurück für die angegebene Menge und den aktuellen User.		
Returns	<code>SET_OBJ_NOT_FOUND</code>	(-3)	Bei einem Fehler.
	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.
Notes	Das SETSet-Objekt muss vorher mit <code>Init()</code> Initialisiert worden sein.		

4.15 CheckSubSet()

Interface	<code>long CheckSubSet(long set_id, long FAR* result)</code>						
Parameters	<p><code>set_id</code> - ID der Untermenge. <code>result</code> - Platzhalter, in welchen das Resultat geschrieben wird:</p> <ul style="list-style-type: none"> • SET_IS_SUBSET (Ist Untermenge), • SET_IS_NO_SUBSET (Ist keine Untermenge). 						
Description	Mit dieser Methode kann überprüft werden, ob die angegebene Menge eine Untermenge von der aktuellen Menge ist.						
Returns	<table> <tr> <td>SET_FAILURE</td> <td>(-14)</td> <td>Bei einem Fehler.</td> </tr> <tr> <td>SET_NO_ERROR</td> <td>(0)</td> <td>Bei keinem Fehler</td> </tr> </table>	SET_FAILURE	(-14)	Bei einem Fehler.	SET_NO_ERROR	(0)	Bei keinem Fehler
SET_FAILURE	(-14)	Bei einem Fehler.					
SET_NO_ERROR	(0)	Bei keinem Fehler					
Notes	Das SETSet-Objekt muss vorher mit <code>Init()</code> Initialisiert worden sein.						

4.16 InitNew()

Interface	<code>long InitNew(BSTR FAR* name)</code>						
Parameters	<code>name</code> - Ein Zeiger auf den Namen der neuen Menge. Ist dieser Name NULL, so wird der eindeutige Name anhand einer Sequenz aus der Datenbank gelesen.						
Description	Diese Methode erzeugt eine Menge mit dem angegebenen Namen. Der Status der Menge ist auf Initialisiert geSetzt.						
Returns	<table> <tr> <td>SET_OBJ_NO_CREATED</td> <td>(-5)</td> <td>Bei einem Fehler.</td> </tr> <tr> <td>SET_NO_ERROR</td> <td>(0)</td> <td>Bei keinem Fehler.</td> </tr> </table>	SET_OBJ_NO_CREATED	(-5)	Bei einem Fehler.	SET_NO_ERROR	(0)	Bei keinem Fehler.
SET_OBJ_NO_CREATED	(-5)	Bei einem Fehler.					
SET_NO_ERROR	(0)	Bei keinem Fehler.					
Notes	Nach diesem Aufruf muss <code>InsertElements()</code> , <code>InsertSingleElement()</code> , <code>InsertMultipleElements()</code> , <code>Copy()</code> oder <code>Merge...()</code> folgen und am Schluss <code>Commit()</code> oder <code>Rollback()</code> .						

4.17 ChangeTypeToRelationSet()

Interface	long ChangeTypeToRelationSet()		
Parameters	-		
Description	Diese Methode wandelt eine Menge, die vorher mit <code>InitNew()</code> erzeugt worden ist, in eine Beziehungsmenge um.		
Returns	SET_NOT_CHANGED	(-12)	Bei einem Fehler.
	SET_NO_ERROR	(0)	Bei keinem Fehler.
Notes	Die Menge muss vorher mit <code>InitNew()</code> erzeugt worden sein. Wenn diese Operation fehl schlägt muss <code>Rollback()</code> aufgerufen werden, ansonsten <code>Commit()</code> .		

4.18 ChangeTypeToAttributeSet()

Interface	long ChangeTypeAttributeSet()		
Parameters	-		
Description	Diese Methode wandelt eine Menge, die vorher mit <code>InitNew()</code> erzeugt worden ist, in eine Attributmenge um.		
Returns	SET_NOT_CHANGED	(-12)	Bei einem Fehler.
	SET_NO_ERROR	(0)	Bei keinem Fehler.
Notes	Die Menge muss vorher mit <code>InitNew()</code> erzeugt worden sein. Wenn diese Operation fehl schlägt muss <code>Rollback()</code> aufgerufen werden, ansonsten <code>Commit()</code> .		

4.19 InsertElements()

Interface	<code>long InsertElements(????? Get_ele_ids)</code>						
Parameters	<code>Get_ele_ids</code> - Funktion welche alle Element-ID's liefert, die gespeichert werden sollen.						
Description	Diese Methode fügt Elemente in die angegebene Menge ein. Bei diesem Funktionsaufruf muss eine callback-Funktion angegeben werden, die die Element-IDs liefert. Der Status der Menge wird automatisch auf IDs (2) geSetzt.						
Returns	<table> <tr> <td><code>SET_NO_INSERT</code></td> <td><code>(-13)</code></td> <td>Bei einem Fehler.</td> </tr> <tr> <td><code>SET_NO_ERROR</code></td> <td><code>(0)</code></td> <td>Bei keinem Fehler.</td> </tr> </table>	<code>SET_NO_INSERT</code>	<code>(-13)</code>	Bei einem Fehler.	<code>SET_NO_ERROR</code>	<code>(0)</code>	Bei keinem Fehler.
<code>SET_NO_INSERT</code>	<code>(-13)</code>	Bei einem Fehler.					
<code>SET_NO_ERROR</code>	<code>(0)</code>	Bei keinem Fehler.					
Notes	Die Menge muss vorher mit <code>InitNew()</code> erzeugt worden sein. Wenn diese Operation fehl schlägt muss <code>Rollback()</code> aufgerufen werden, ansonsten <code>Commit()</code> .						

4.20 InsertSingleElement()

Interface	<code>long InsertSingleElement(BSTR FAR* values)</code>						
Parameters	<code>values</code> - String für die Werte, der verschiedenen Spalten, in der Menge (z.B. " 99, 'Muster' ").						
Description	Diese Methode fügt ein Element mit den notwendigen Attribute in die angegebene Menge ein. Die Menge darf eine Objektmenge, Attributmenge, Beziehungsmenge oder ein Infopaket sein.						
Returns	<table> <tr> <td><code>SET_NO_INSERT</code></td> <td><code>(-13)</code></td> <td>Bei einem Fehler.</td> </tr> <tr> <td><code>SET_NO_ERROR</code></td> <td><code>(0)</code></td> <td>Bei keinem Fehler.</td> </tr> </table>	<code>SET_NO_INSERT</code>	<code>(-13)</code>	Bei einem Fehler.	<code>SET_NO_ERROR</code>	<code>(0)</code>	Bei keinem Fehler.
<code>SET_NO_INSERT</code>	<code>(-13)</code>	Bei einem Fehler.					
<code>SET_NO_ERROR</code>	<code>(0)</code>	Bei keinem Fehler.					
Notes	Die Menge muss vorher mit <code>InitNew()</code> erzeugt oder mit <code>Lock()</code> gesperrt worden sein. Wenn diese Operation fehl schlägt muss <code>Rollback()</code> aufgerufen werden, ansonsten <code>Commit()</code> .						

4.21 InsertMultipleElements()

Interface	<code>long InsertMultipleElements(BSTR FAR* sql_stmt)</code>						
Parameters	<code>sql_stmt</code> - String der ein SQL-Statement beinhaltet.						
Description	Diese Methode fügt Element mittels einem 'select' SQL-Statement in die angegebene Menge ein. Die Menge darf eine Objektmenge, Attributmenge, Beziehungsmenge oder ein Infopaket sein.						
Returns	<table> <tr> <td><code>SET_NO_INSERT</code></td> <td><code>(-13)</code></td> <td>Bei einem Fehler.</td> </tr> <tr> <td><code>SET_NO_ERROR</code></td> <td><code>(0)</code></td> <td>Bei keinem Fehler.</td> </tr> </table>	<code>SET_NO_INSERT</code>	<code>(-13)</code>	Bei einem Fehler.	<code>SET_NO_ERROR</code>	<code>(0)</code>	Bei keinem Fehler.
<code>SET_NO_INSERT</code>	<code>(-13)</code>	Bei einem Fehler.					
<code>SET_NO_ERROR</code>	<code>(0)</code>	Bei keinem Fehler.					
Notes	<p>Die Menge muss vorher mit <code>InitNew()</code> erzeugt oder mit <code>Lock()</code> gesperrt worden sein.</p> <p>Wenn diese Operation fehl schlägt muss <code>Rollback()</code> aufgerufen werden, ansonsten <code>Commit()</code>.</p>						

4.22 DeleteElement()

Interface	<code>long DeleteElement(long FAR* element_id)</code>						
Parameters	<code>element_id</code> - ID des Elementes, welches gelöscht werden soll.						
Description	Diese Methode löscht ein Element in der Menge. Die Menge darf eine Objektmenge, Attributmenge, Beziehungsmenge oder ein Infopaket sein.						
Returns	<table> <tr> <td><code>SET_NO_FAILURE</code></td> <td><code>(-14)</code></td> <td>Bei einem Fehler.</td> </tr> <tr> <td><code>SET_NO_ERROR</code></td> <td><code>(0)</code></td> <td>Bei keinem Fehler.</td> </tr> </table>	<code>SET_NO_FAILURE</code>	<code>(-14)</code>	Bei einem Fehler.	<code>SET_NO_ERROR</code>	<code>(0)</code>	Bei keinem Fehler.
<code>SET_NO_FAILURE</code>	<code>(-14)</code>	Bei einem Fehler.					
<code>SET_NO_ERROR</code>	<code>(0)</code>	Bei keinem Fehler.					
Notes	<p>Die Menge muss vorher mit <code>Lock()</code> gesperrt worden sein.</p> <p>Wenn diese Operation fehl schlägt muss <code>Rollback()</code> aufgerufen werden, ansonsten <code>Commit()</code>.</p>						

4.23 DeleteMultipleElements()

Interface	<code>long DeleteMultipleElements(BSTR FAR* sql_stmt)</code>		
Parameters	<code>sql_stmt</code> - String der ein SQL-Statement beinhaltet.		
Description	Diese Methode löscht Elemente mittels einem 'select' SQL-Statement in die Menge. Die Menge darf eine Objektmenge, Attributmenge, Beziehungsmenge oder ein Infopaket sein.		
Returns	<code>SET_NO_FAILURE</code>	(-14)	Bei einem Fehler.
	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.
Notes	Die Menge muss vorher mit <code>Lock()</code> gesperrt worden sein. Wenn diese Operation fehl schlägt muss <code>Rollback()</code> aufgerufen werden, ansonsten <code>Commit()</code> .		

4.24 Copy ()

Interface	<code>long Copy(long source_set_id)</code>		
Parameters	<code>source_set_id</code> - ID der Quell-Menge, die kopiert werden soll.		
Description	Diese Methode kopiert eine Menge.		
Returns	<code>SET_FAILURE</code>	(-14)	Bei einem Fehler.
	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.
Notes	Die Ziel-Menge muss vorher mit <code>InitNew()</code> erzeugt worden sein. Wenn diese Operation fehl schlägt muss <code>Rollback()</code> aufgerufen werden, ansonsten <code>Commit()</code> .		

4.25 Transform ()

Interface	<code>long transform(long source_set_id)</code>						
Parameters	<code>source_set_id</code> - ID der Beziehungsmenge, die in eine Objektmenge transformiert wird.						
Description	Mit dieser Methode wird die 'nach'-Spalte einer Beziehungsmenge in die Menge überführt.						
Returns	<table> <tr> <td><code>SET_FAILURE</code></td> <td>(-14)</td> <td>Bei einem Fehler.</td> </tr> <tr> <td><code>SET_NO_ERROR</code></td> <td>(0)</td> <td>Bei keinem Fehler.</td> </tr> </table>	<code>SET_FAILURE</code>	(-14)	Bei einem Fehler.	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.
<code>SET_FAILURE</code>	(-14)	Bei einem Fehler.					
<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.					
Notes	<p>Die Menge muss vorher mit <code>InitNew()</code> erzeugt worden sein. Wenn die Funktion erfolgreich war, muss mittels den Funktionen <code>SetScope()</code> und <code>SetDescription()</code> der Besitzstand bzw. die Beschreibung gesetzt werden. Danach muss mit <code>Commit()</code> die Transaktion beendet oder mit <code>Rollback()</code> zurückgesetzt werden.</p>						

4.26 MergeAnd()

Interface	<code>long MergeAnd(long set_id_1, long set_id_2, long with_attrs, long FAR* newRows)</code>						
Parameters	<p><code>set_id_1</code> - ID der ersten Quell-Menge. <code>set_id_2</code> - ID der zweiten Quell-Menge. <code>with_attrs</code> - Flag, ob bei der ersten Menge die Attribute mit kopiert werden sollen. <code>newRows</code> - Platzhalter, in welchen die Anzahl der Zeilen, die in die Menge eingetragen worden sind, geschrieben wird.</p>						
Description	Diese Methode bildet die Schnittmenge von zwei Mengen der selben Entität.						
Returns	<table> <tr> <td><code>SET_FAILURE</code></td> <td>(-14)</td> <td>Bei einem Fehler.</td> </tr> <tr> <td><code>SET_NO_ERROR</code></td> <td>(0)</td> <td>Bei keinem Fehler.</td> </tr> </table>	<code>SET_FAILURE</code>	(-14)	Bei einem Fehler.	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.
<code>SET_FAILURE</code>	(-14)	Bei einem Fehler.					
<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.					
Notes	<p>Die Ziel-Menge muss vorher mit <code>InitNew()</code> erzeugt worden sein. Wenn diese Operation fehl schlägt muss <code>Rollback()</code> aufgerufen werden, ansonsten <code>Commit()</code>.</p>						

4.27 MergeOr()

Interface	<code>long MergeOr(long set_id_1, long set_id_2, long with_attrs, long FAR* newRows)</code>
Parameters	<p><code>set_id_1</code> - ID der ersten Quell-Menge. <code>set_id_2</code> - ID der zweiten Quell-Menge. <code>with_attrs</code> - Flag, ob bei der ersten Menge die Attribute mit kopiert werden sollen. <code>newRows</code> - Platzhalter, in welchen die Anzahl der Zeilen, die in die Menge einGetragen worden sind, geschrieben wird.</p>
Description	Diese Methode bildet die Vereinigungsmenge von zwei Mengen der selben Entität.
Returns	<p><code>SET_FAILURE</code> (-14) Bei einem Fehler. <code>SET_NO_ERROR</code> (0) Bei keinem Fehler.</p>
Notes	<p>Die Ziel-Menge muss vorher mit <code>InitNew()</code> erzeugt worden sein. Wenn diese Operation fehl schlägt muss <code>Rollback()</code> aufgerufen werden, ansonsten <code>Commit()</code>.</p>

4.28 MergeMinus()

Interface	<code>long MergeMinus(long set_id_1, long set_id_2, long with_attrs, long FAR* newRows)</code>						
Parameters	<p><code>set_id_1</code> - ID der ersten Quell-Menge. <code>set_id_2</code> - ID der zweiten Quell-Menge. <code>with_attrs</code> - Flag, ob bei der ersten Menge die Attribute mit kopiert werden sollen. <code>newRows</code> - Platzhalter, in welchen die Anzahl der Zeilen, die in die Menge einGetragen worden sind, geschrieben wird.</p>						
Description	Diese Methode bildet die Differenzmenge von zwei Mengen der selben Entität.						
Returns	<table> <tr> <td><code>SET_FAILURE</code></td> <td>(-14)</td> <td>Bei einem Fehler.</td> </tr> <tr> <td><code>SET_NO_ERROR</code></td> <td>(0)</td> <td>Bei keinem Fehler.</td> </tr> </table>	<code>SET_FAILURE</code>	(-14)	Bei einem Fehler.	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.
<code>SET_FAILURE</code>	(-14)	Bei einem Fehler.					
<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.					
Notes	<p>Die Ziel-Menge muss vorher mit <code>InitNew()</code> erzeugt worden sein. Wenn diese Operation fehl schlägt muss <code>Rollback()</code> aufgerufen werden, ansonsten <code>Commit()</code>.</p>						

4.29 Commit()

Interface	<code>long Commit()</code>						
Parameters	-						
Description	<p>Diese Methode beendet die Erzeugung einer neuen Menge, wenn zuvor <code>InitNew()</code> in Kombination mit <code>InsertElements()</code>, <code>InsertSingleElement()</code>, <code>InsertMultipleElements()</code>, <code>Copy()</code> oder <code>Merge...()</code> aufgerufen worden ist oder die Methode beendet die Manipulation der Zustandsänderung der Menge (<code>SetName()</code>, <code>SetEntity()</code>, ...), sofern die Menge vorgängig mit <code>Lock()</code> gesperrt worden ist.</p> <p>Ebenfalls wird diese Methode bei der Manipulation von Attributen (siehe <code>SETAttribute</code>) benötigt. Die Methode beendet die Erzeugung eines neuen Attributes, wenn zuvor <code>InitNewRbs()</code>, <code>InitNewArithmetic()</code> oder <code>InitNew...()</code> (Aggregatfunktionen) aufgerufen worden ist oder die Methode ändert ein Attribut in der Datenbank, wenn dieses zuvor mit <code>SetName()</code> oder <code>SetDescription()</code> editiert worden ist.</p>						
Returns	<table> <tr> <td><code>SET_FAILURE</code></td> <td>(-14)</td> <td>Bei einem Fehler.</td> </tr> <tr> <td><code>SET_NO_ERROR</code></td> <td>(0)</td> <td>Bei keinem Fehler.</td> </tr> </table>	<code>SET_FAILURE</code>	(-14)	Bei einem Fehler.	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.
<code>SET_FAILURE</code>	(-14)	Bei einem Fehler.					
<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.					
Notes	<p>Bei der Verwendung dieser Methode von Mengenmanipulationen müssen, bevor diese Methode aufgerufen werden kann, mindestens die Entitäts- und die Projektzugehörigkeit mit <code>SetEntity()/SetProject()</code> gesetzt werden. Die anderen Komponenten werden automatisch oder mit Default-Werten gefüllt.</p> <p>Wenn diese Operation fehl schlägt muss <code>Rollback()</code> aufgerufen werden.</p>						

4.30 Rollback ()

Interface	long Rollback()						
Parameters	-						
Description	<p>Diese Methode beendet die vorgängige(n) Operation(en). Sie darf nach <code>InitNew()</code>, <code>InsertElements()</code>, <code>Lock()</code>, <code>InsertMultipleElements()</code>, <code>InsertSingleElement()</code>, <code>Copy()</code> oder <code>Merge...()</code> aufgerufen werden.</p> <p>Ebenfalls wird diese Methode bei der Manipulation von Attributen (siehe <code>SETAttribute</code>) benötigt. Die Methode die Methode darf nach <code>InitNewRbs()</code>, <code>InitNewArithmetic()</code> oder <code>InitNew...()</code> (Aggregatfunktionen) aufgerufen werden.</p>						
Returns	<table> <tr> <td><code>SET_FAILURE</code></td> <td>(-14)</td> <td>Bei einem Fehler.</td> </tr> <tr> <td><code>SET_NO_ERROR</code></td> <td>(0)</td> <td>Bei keinem Fehler.</td> </tr> </table>	<code>SET_FAILURE</code>	(-14)	Bei einem Fehler.	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.
<code>SET_FAILURE</code>	(-14)	Bei einem Fehler.					
<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.					

4.31 Lock ()

Interface	long Lock()						
Parameters	-						
Description	<p>Diese Methode sperrt die Menge für Änderungszwecke.</p> <p>Ebenfalls wird diese Methode verwendet, um die Menge zu sperren, wenn Manipulationen an Attributen erfolgt.</p>						
Returns	<table> <tr> <td><code>NO_FAILURE</code></td> <td>(-14)</td> <td>Bei einem Fehler.</td> </tr> <tr> <td><code>NO_ERROR</code></td> <td>(0)</td> <td>Bei keinem Fehler.</td> </tr> </table>	<code>NO_FAILURE</code>	(-14)	Bei einem Fehler.	<code>NO_ERROR</code>	(0)	Bei keinem Fehler.
<code>NO_FAILURE</code>	(-14)	Bei einem Fehler.					
<code>NO_ERROR</code>	(0)	Bei keinem Fehler.					
Notes	Nach diesem Funktionsaufruf muss <code>Commit()</code> oder <code>Rollback()</code> folgen.						

4.32 SetScope()

Interface	<code>long SetScope(long FAR* value)</code>						
Parameters	<code>value</code> - Wert, mit dem der Besitzstand überschrieben werden soll.						
Description	Mit dieser Funktion kann der Besitzstand der Menge geändert oder erfasst werden (Privat =0, Public =1).						
Returns	<table> <tr> <td><code>SET_FAILURE</code></td> <td>(-14)</td> <td>Bei einem Fehler.</td> </tr> <tr> <td><code>SET_NO_ERROR</code></td> <td>(0)</td> <td>Bei keinem Fehler.</td> </tr> </table>	<code>SET_FAILURE</code>	(-14)	Bei einem Fehler.	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.
<code>SET_FAILURE</code>	(-14)	Bei einem Fehler.					
<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.					
Notes	Die Menge muss vorher mit <code>Lock()</code> gesperrt oder mit <code>InitNew()</code> neu angelegt werden.						

4.33 SetEntity()

Interface	<code>long SetEntity(long FAR* value)</code>						
Parameters	<code>value</code> - Wert, mit dem die Entitätszugehörigkeit überschrieben werden soll.						
Description	Mit dieser Funktion kann die Entitätszugehörigkeit der Menge geändert oder erfasst werden (ID der Entität).						
Returns	<table> <tr> <td><code>SET_FAILURE</code></td> <td>(-14)</td> <td>Bei einem Fehler.</td> </tr> <tr> <td><code>SET_NO_ERROR</code></td> <td>(0)</td> <td>Bei keinem Fehler.</td> </tr> </table>	<code>SET_FAILURE</code>	(-14)	Bei einem Fehler.	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.
<code>SET_FAILURE</code>	(-14)	Bei einem Fehler.					
<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.					
Notes	Die Menge muss vorher mit <code>Lock()</code> gesperrt oder mit <code>InitNew()</code> neu angelegt werden.						

4.34 SetProject()

Interface	<code>long SetProject(long FAR* value)</code>						
Parameters	<code>value</code> - Wert, mit dem die Projektzugehörigkeit überschrieben werden soll.						
Description	Mit dieser Funktion kann die Projektzugehörigkeit der Menge geändert oder erfasst werden (ID der langen Transaktion).						
Returns	<table> <tr> <td><code>SET_FAILURE</code></td> <td>(-14)</td> <td>Bei einem Fehler.</td> </tr> <tr> <td><code>SET_NO_ERROR</code></td> <td>(0)</td> <td>Bei keinem Fehler.</td> </tr> </table>	<code>SET_FAILURE</code>	(-14)	Bei einem Fehler.	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.
<code>SET_FAILURE</code>	(-14)	Bei einem Fehler.					
<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.					
Notes	Die Menge muss vorher mit <code>Lock()</code> gesperrt oder mit <code>InitNew()</code> neu angelegt werden.						

4.35 SetName()

Interface	<code>long SetName(BSTR FAR* value)</code>						
Parameters	<code>value</code> - Wert, mit dem der Name der Menge überschrieben werden soll.						
Description	Mit dieser Methode kann der Name der Menge geändert oder erfasst werden.						
Returns	<table> <tr> <td><code>SET_FAILURE</code></td> <td>(-14)</td> <td>Bei einem Fehler.</td> </tr> <tr> <td><code>SET_NO_ERROR</code></td> <td>(0)</td> <td>Bei keinem Fehler.</td> </tr> </table>	<code>SET_FAILURE</code>	(-14)	Bei einem Fehler.	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.
<code>SET_FAILURE</code>	(-14)	Bei einem Fehler.					
<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.					
Notes	Die Menge muss vorher mit <code>Lock()</code> gesperrt oder mit <code>InitNew()</code> neu angelegt werden.						

4.36 SetDescription()

Interface	<code>long SetDescription(BSTR FAR* value)</code>		
Parameters	<code>value</code> - Wert, mit dem die Beschreibung der Menge überschrieben werden soll.		
Description	Mit dieser Methode kann die Beschreibung der Menge geändert oder erfasst werden.		
Returns	<code>SET_FAILURE</code>	(-14)	Bei einem Fehler.
	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.
Notes	Die Menge muss vorher mit <code>Lock()</code> gesperrt oder mit <code>InitNew()</code> neu angelegt werden.		

4.37 SetNameRef()

Interface	<code>long SetNameRef()</code>		
Parameters	-		
Description	Diese Methode ändert den Status der Menge (3: Referenziert durch Elementname) und kopiert automatisch den sprechenden RBS-Identifikator - OBJEKTNAME an die bestehende Menge, sofern diese den Status 'Referenziert durch ID' hat.		
Returns	<code>SET_FAILURE</code>	(-14)	Bei einem Fehler.
	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.
Notes	Die Menge muss vorher mit <code>Lock()</code> gesperrt oder mit <code>InitNew()</code> neu erzeugt worden sein. Wenn diese Operation fehl schlägt muss <code>Rollback()</code> aufgerufen werden, ansonsten <code>Commit()</code> .		

4.38 Delete ()

Interface	long Delete()		
Parameters	-		
Description	Mit dieser Methode kann die Menge gelöscht werden.		
Returns	SET_FAILURE	(-14)	Bei einem Fehler.
	SET_NO_ERROR	(0)	Bei keinem Fehler.
Notes	Das SETSet-Objekt muss vorher mit <code>Init()</code> Initialisiert worden sein.		

4.39 Reset()

Interface	long Reset()		
Parameters	-		
Description	Diese Methode Setzt alle vorgängigen, lokalen Operationen zurück.		
Returns	SET_FAILURE	(-14)	Bei einem Fehler.
	SET_NO_ERROR	(0)	Bei keinem Fehler.
Notes	Das SETSet-Objekt muss vorher mit <code>Init()</code> Initialisiert worden sein.		

4.40 GetFirstElement()

Interface	<code>long GetFirstElement(long FAR* element_id)</code>		
Parameters	<code>element_id</code> - Platzhalter, in welchen die erste selektierte Element-Id geschrieben wird.		
Description	Mit dieser Methode wird die erste Element-ID in der Menge zurückgegeben.		
Returns	<code>SET_FAILURE</code>	(-14)	Bei einem Fehler.
	<code>SET_EOF</code>	(1)	End of table
	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.
Notes	Das SETSet-Objekt muss vorher mit <code>Init()</code> Initialisiert worden sein.		

4.41 GetNextElement()

Interface	<code>long GetNextElement(long FAR* element_id)</code>		
Parameters	<code>element_id</code> - Platzhalter, in welchen die Element-Id geschrieben wird.		
Description	Mit dieser Methode wird jeweils die nächste Element-ID in der Menge zurückgegeben.		
Returns	<code>SET_FAILURE</code>	(-14)	Bei einem Fehler.
	<code>SET_EOF</code>	(1)	End of table
	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.
Notes	Das SETSet-Objekt muss mit <code>Init()</code> Initialisiert und die Methode <code>GetFirstElement()</code> aufgerufen worden sein. Der Aufruf erfolgt in einer Schleife, bis der Rückgabewert gleich <code>SET_EOF</code> ist.		

4.42 CreateAttributeList()

Interface	long CreateAttributeList()		
Parameters	-		
Description	Mit dieser Methode werden über ODBC die Attribute der Menge aus Verwaltungstabelle gelesen und im Memory abgelegt. Mit den Methoden <code>GetFirstAttribute()</code> und <code>GetNextAttribute()</code> kann der Name der Menge, in der aktuellen Liste, gelesen werden.		
Returns	SET_NO_DB_CONNECT	(-9)	Keine Verbindung zur Datenbank.
	DB_SQL_ERROR	(-2)	Internes SQL-Statement ist falsch.
	SET_NO_ERROR	(0)	Bei keinem Fehler.
Notes	Das SETSet-Objekt muss vorher mit <code>Init()</code> Initialisiert worden sein.		

4.43 FreeAttributeList()

Interface	void FreeAttributeList()		
Parameters	-		
Description	Diese Methode gibt den Speicher frei, der in der Methode <code>CreateAttributeList()</code> alloziert worden ist.		
Returns	-		
Notes	Vorgängig muss die Funktion <code>CreateAttributeList()</code> aufgerufen worden sein.		

4.44 GetFirstAttribute()

Interface	long GetFirstAttribute(BSTR FAR* name)		
Parameters	name - Platzhalter, in welchen der erste Attributname der Liste geschrieben wird.		
Description	Mit dieser Methode wird der erste Attributname der Liste zurückgegeben.		
Returns	SET_LIST_NOT_INIT	(-1)	Die Liste ist vorgängig nicht erzeugt worden.
	SET_OBJ_NOT_FOUND	(-3)	Kein Objekt gefunden.
	SET_NO_ERROR	(0)	Bei keinem Fehler.
Notes	Die Methode CreateAttributeList() muss vorgängig aufgerufen worden sein.		

4.45 GetNextAttribute()

Interface	long GetNext(BSTR FAR* name)		
Parameters	name - Platzhalter, in welchen der Attributname der Liste geschrieben wird.		
Description	Mit dieser Methode wird jeweils der nächste Attributname der Liste zurückgegeben.		
Returns	SET_LIST_NOT_INIT	(-1)	Die Liste ist vorgängig nicht erzeugt worden.
	SET_EOF	(1)	Listenende erreicht..
	SET_NO_ERROR	(0)	Bei keinem Fehler.
Notes	Die Methode CreateAttributeList() und GetFirstAttribute() muss vorgängig aufgerufen worden sein.		

5 Methoden für Attribut-Objekt (SETAttribute)

5.1 Init()

Interface	<code>long Init(long pos)</code>						
Parameters	<code>pos</code> - Positionsindex des Attributs in der Liste.						
Description	Diese Methode Initialisiert, gemäss dem Positionsindex in der Liste, ein Objekt der Klasse SETAttribute.						
Returns	<table> <tr> <td>SET_FAILURE</td> <td>(-14)</td> <td>Bei einem Fehler.</td> </tr> <tr> <td>SET_NO_ERROR</td> <td>(0)</td> <td>Bei keinem Fehler.</td> </tr> </table>	SET_FAILURE	(-14)	Bei einem Fehler.	SET_NO_ERROR	(0)	Bei keinem Fehler.
SET_FAILURE	(-14)	Bei einem Fehler.					
SET_NO_ERROR	(0)	Bei keinem Fehler.					
Notes	Die Liste der Attribute muss vorher mit <code>SETSet.CreateAttributeList()</code> erzeugt worden sein.						

5.2 InitNewRbs()

Interface	<code>long InitNewRbs(long attr_id, BSTR FAR* name)</code>						
Parameters	<code>attr_id</code> - ID des Attributs, welches im SDD gespeichert ist. <code>name</code> - Name des Attributes (kann unterschiedlich sein, zum SDD-Attributnamen).						
Description	Mit dieser Methode kann ein RBS-Attribut (<u>R</u> aum <u>b</u> ezug <u>s</u> ystem-Attribut) an eine Menge kopiert werden.						
Returns	<table> <tr> <td>SET_FAILURE</td> <td>(-14)</td> <td>Bei einem Fehler.</td> </tr> <tr> <td>SET_NO_ERROR</td> <td>(0)</td> <td>Bei keinem Fehler.</td> </tr> </table>	SET_FAILURE	(-14)	Bei einem Fehler.	SET_NO_ERROR	(0)	Bei keinem Fehler.
SET_FAILURE	(-14)	Bei einem Fehler.					
SET_NO_ERROR	(0)	Bei keinem Fehler.					
Notes	Wenn diese Operation fehl schlägt muss <code>Rollback()</code> aufgerufen werden, ansonsten <code>Commit()</code> .						

5.3 InitNewArithmetic()

Interface	<code>long InitNewArithmetic(long formula_id, ???? attr_pos_array, long no, BSTR FAR* attr_name)</code>
Parameters	<p><code>formula_id</code> - ID der arithmetischen Formel mit welcher das neue Attribut berechnet und erzeugt wird.</p> <p><code>attr_pos_array</code> - Liste von Attribut-Positionsindizes</p> <p><code>no</code> - Anzahl der Attribute die benötigt werden.</p> <p><code>attr_name</code> - Name des neuen Attributes welches automatisch erstellt wird.</p>
Description	Mit dieser Methode kann mittels einer arithmetische Formel, die im SDD gespeichert ist und über die Number-Attribute der aktuellen Menge ein neues Attribut erzeugt werden.
Returns	<p><code>SET_FAILURE</code> (-14) Bei einem Fehler.</p> <p><code>SET_NO_ERROR</code> (0) Bei keinem Fehler.</p>
Notes	Wenn diese Operation fehl schlägt muss <code>Rollback()</code> aufgerufen werden, ansonsten <code>Commit()</code> .

5.4 InitNewAverage()

Interface	<code>long InitNewAverage(long set_id_1, long set_id_2, long attr_id, BSTR FAR* attr_name)</code>
Parameters	<p><code>set_id_1</code> - ID der Attributmenge. <code>set_id_2</code> - ID der Beziehungsmenge. <code>attr_id</code> - ID des Attributes der Attributmenge. <code>attr_name</code> - Name des neuen Attributes welches automatisch erstellt wird.</p>
Description	Mit dieser Funktion kann ein Attribut einer Attributmenge über eine Beziehungsmenge aggregiert werden. Die Aggregation wird mittels der Funktion 'Durchschnitt' berechnet.
Returns	<p><code>SET_FAILURE</code> (-14) Bei einem Fehler. <code>SET_NO_ERROR</code> (0) Bei keinem Fehler.</p>
Notes	Wenn diese Operation fehl schlägt muss <code>Rollback()</code> aufgerufen werden, ansonsten <code>Commit()</code> .

5.5 InitNewSum()

Interface	<code>long InitNewSum(long set_id_1, long set_id_2, long attr_id, BSTR FAR* attr_name)</code>
Parameters	<p><code>set_id_1</code> - ID der Attributmenge. <code>set_id_2</code> - ID der Beziehungsmenge. <code>attr_id</code> - ID des Attributes der Attributmenge. <code>attr_name</code> - Name des neuen Attributes welches automatisch erstellt wird.</p>
Description	Mit dieser Funktion kann ein Attribut einer Attributmenge über eine Beziehungsmenge aggregiert werden. Die Aggregation wird mittels der Funktion 'Summe' berechnet.
Returns	<p><code>SET_FAILURE</code> (-14) Bei einem Fehler. <code>SET_NO_ERROR</code> (0) Bei keinem Fehler.</p>
Notes	Wenn diese Operation fehl schlägt muss <code>Rollback()</code> aufgerufen werden, ansonsten <code>Commit()</code> .

5.6 InitNewMaximum()

Interface	<code>long InitNewMaximum(long set_id_1, long set_id_2, long attr_id, BSTR FAR* attr_name)</code>						
Parameters	<p><code>set_id_1</code> - ID der Attributmenge. <code>set_id_2</code> - ID der Beziehungsmenge. <code>attr_id</code> - ID des Attributes der Attributmenge. <code>attr_name</code> - Name des neuen Attributes welches automatisch erstellt wird.</p>						
Description	Mit dieser Funktion kann ein Attribut einer Attributmenge über eine Beziehungsmenge aggregiert werden. Die Aggregation wird mittels der Funktion 'Maximum' berechnet.						
Returns	<table> <tr> <td><code>SET_FAILURE</code></td> <td>(-14)</td> <td>Bei einem Fehler.</td> </tr> <tr> <td><code>SET_NO_ERROR</code></td> <td>(0)</td> <td>Bei keinem Fehler.</td> </tr> </table>	<code>SET_FAILURE</code>	(-14)	Bei einem Fehler.	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.
<code>SET_FAILURE</code>	(-14)	Bei einem Fehler.					
<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.					
Notes	Wenn diese Operation fehl schlägt muss <code>Rollback()</code> aufgerufen werden, ansonsten <code>Commit()</code> .						

5.7 InitNewMinimum()

Interface	<code>long InitNewMinimum(long set_id_1, long set_id_2, long attr_id, BSTR FAR* attr_name)</code>						
Parameters	<p><code>set_id_1</code> - ID der Attributmenge. <code>set_id_2</code> - ID der Beziehungsmenge. <code>attr_id</code> - ID des Attributes der Attributmenge. <code>attr_name</code> - Name des neuen Attributes welches automatisch erstellt wird.</p>						
Description	Mit dieser Funktion kann ein Attribut einer Attributmenge über eine Beziehungsmenge aggregiert werden. Die Aggregation wird mittels der Funktion 'Minimum' berechnet.						
Returns	<table> <tr> <td><code>SET_FAILURE</code></td> <td>(-14)</td> <td>Bei einem Fehler.</td> </tr> <tr> <td><code>SET_NO_ERROR</code></td> <td>(0)</td> <td>Bei keinem Fehler.</td> </tr> </table>	<code>SET_FAILURE</code>	(-14)	Bei einem Fehler.	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.
<code>SET_FAILURE</code>	(-14)	Bei einem Fehler.					
<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.					
Notes	Wenn diese Operation fehl schlägt muss <code>Rollback()</code> aufgerufen werden, ansonsten <code>Commit()</code> .						

5.8 InitNewCount()

Interface	<code>long InitNewCount(long set_id_1, long set_id_2, long attr_id, BSTR FAR* attr_name)</code>
Parameters	<p><code>set_id_1</code> - ID der Attributmenge. <code>set_id_2</code> - ID der Beziehungsmenge. <code>attr_id</code> - ID des Attributes der Attributmenge. <code>attr_name</code> - Name des neuen Attributes welches automatisch erstellt wird.</p>
Description	Mit dieser Funktion kann ein Attribut einer Attributmenge über eine Beziehungsmenge aggregiert werden. Die Aggregation wird mittels der Funktion 'Anzahl' berechnet.
Returns	<p><code>SET_FAILURE</code> (-14) Bei einem Fehler. <code>SET_NO_ERROR</code> (0) Bei keinem Fehler.</p>
Notes	Wenn diese Operation fehl schlägt muss <code>Rollback()</code> aufgerufen werden, ansonsten <code>Commit()</code> .

5.9 InitNewStddev()

Interface	<code>long InitNewStddev(long set_id_1, long set_id_2, long attr_id, BSTR FAR* attr_name)</code>
Parameters	<p><code>set_id_1</code> - ID der Attributmenge. <code>set_id_2</code> - ID der Beziehungsmenge. <code>attr_id</code> - ID des Attributes der Attributmenge. <code>attr_name</code> - Name des neuen Attributes welches automatisch erstellt wird.</p>
Description	Mit dieser Funktion kann ein Attribut einer Attributmenge über eine Beziehungsmenge aggregiert werden. Die Aggregation wird mittels der Funktion 'Standardabweichung' berechnet.
Returns	<p><code>SET_FAILURE</code> (-14) Bei einem Fehler. <code>SET_NO_ERROR</code> (0) Bei keinem Fehler.</p>
Notes	Wenn diese Operation fehl schlägt muss <code>Rollback()</code> aufgerufen werden, ansonsten <code>Commit()</code> .

5.10 InitNewVariance()

Interface	<code>long InitNewVariance(long set_id_1, long set_id_2, long attr_id, BSTR FAR* attr_name)</code>						
Parameters	<p><code>set_id_1</code> - ID der Attributmenge. <code>set_id_2</code> - ID der Beziehungsmenge. <code>attr_id</code> - ID des Attributes der Attributmenge. <code>attr_name</code> - Name des neuen Attributes welches automatisch erstellt wird.</p>						
Description	Mit dieser Funktion kann ein Attribut einer Attributmenge über eine Beziehungsmenge aggregiert werden. Die Aggregation wird mittels der Funktion 'Varianz' berechnet.						
Returns	<table> <tr> <td><code>SET_FAILURE</code></td> <td>(-14)</td> <td>Bei einem Fehler.</td> </tr> <tr> <td><code>SET_NO_ERROR</code></td> <td>(0)</td> <td>Bei keinem Fehler.</td> </tr> </table>	<code>SET_FAILURE</code>	(-14)	Bei einem Fehler.	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.
<code>SET_FAILURE</code>	(-14)	Bei einem Fehler.					
<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.					
Notes	Wenn diese Operation fehl schlägt muss <code>Rollback()</code> aufgerufen werden, ansonsten <code>Commit()</code> .						

5.11 SetName()

Interface	<code>long SetName(BSTR FAR* name)</code>						
Parameters	<code>name</code> - Neuer Attributname.						
Description	Mit dieser Methode kann der Attributname geändert werden.						
Returns	<table> <tr> <td><code>SET_FAILURE</code></td> <td>(-14)</td> <td>Bei einem Fehler.</td> </tr> <tr> <td><code>SET_NO_ERROR</code></td> <td>(0)</td> <td>Bei keinem Fehler.</td> </tr> </table>	<code>SET_FAILURE</code>	(-14)	Bei einem Fehler.	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.
<code>SET_FAILURE</code>	(-14)	Bei einem Fehler.					
<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.					
Notes	<p>Das Attribut muss vorher mit <code>Init()</code> initialisiert und mit <code>Lock()</code> gesperrt worden sein. Wenn diese Operation fehl schlägt muss <code>Rollback()</code> aufgerufen werden ansonsten <code>Commit()</code>.</p>						

5.12 setDescription()

Interface	<code>long setDescription(BSTR FAR* name)</code>						
Parameters	<code>name</code> - Neue Beschreibung des Attributes						
Description	Mit dieser Methode kann die Beschreibung des Attributes geändert werden.						
Returns	<table> <tr> <td><code>SET_FAILURE</code></td> <td><code>(-14)</code></td> <td>Bei einem Fehler.</td> </tr> <tr> <td><code>SET_NO_ERROR</code></td> <td><code>(0)</code></td> <td>Bei keinem Fehler.</td> </tr> </table>	<code>SET_FAILURE</code>	<code>(-14)</code>	Bei einem Fehler.	<code>SET_NO_ERROR</code>	<code>(0)</code>	Bei keinem Fehler.
<code>SET_FAILURE</code>	<code>(-14)</code>	Bei einem Fehler.					
<code>SET_NO_ERROR</code>	<code>(0)</code>	Bei keinem Fehler.					
Notes	<p>Das Attribut muss vorher mit <code>Init()</code> Initialisiert und mit <code>Lock()</code> gesperrt worden sein. Wenn diese Operation fehl schlägt muss <code>Rollback()</code> aufgerufen werden.</p> <p>Bei Beendigung aller <code>Set...</code>-Transaktion(en) muss <code>Rollback()</code> oder <code>update()</code> aufgerufen werden.</p>						

5.13 GetType()

Interface	<code>long GetType(long FAR* value)</code>						
Parameters	<code>value</code> - Platzhalter, in welchen der Typ geschrieben wird.						
Description	Mit dieser Methode kann der Typ des Attributes abgefragt werden (Integer, Float, String, Datum).						
Returns	<table> <tr> <td><code>SET_FAILURE</code></td> <td><code>(-14)</code></td> <td>Bei einem Fehler.</td> </tr> <tr> <td><code>SET_NO_ERROR</code></td> <td><code>(0)</code></td> <td>Bei keinem Fehler.</td> </tr> </table>	<code>SET_FAILURE</code>	<code>(-14)</code>	Bei einem Fehler.	<code>SET_NO_ERROR</code>	<code>(0)</code>	Bei keinem Fehler.
<code>SET_FAILURE</code>	<code>(-14)</code>	Bei einem Fehler.					
<code>SET_NO_ERROR</code>	<code>(0)</code>	Bei keinem Fehler.					
Notes	Das SETAttribute-Objekt muss vorher mit <code>Init()</code> Initialisiert worden sein.						

5.14 GetLength()

Interface	<code>long GetLength(long FAR* value)</code>		
Parameters	<code>value</code> - Platzhalter, in welchen der Länge des Attributfeldes geschrieben wird.		
Description	Mit dieser Methode kann die Länge des Attributfeldes abgefragt werden.		
Returns	<code>SET_FAILURE</code>	(-14)	Bei einem Fehler.
	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.
Notes	Das SETAttribute-Objekt muss vorher mit <code>Init()</code> Initialisiert worden sein.		

5.15 GetPercision()

Interface	<code>long GetPercision(long FAR* value)</code>		
Parameters	<code>value</code> - Platzhalter, in welchen die Anzahl Nachkommastellen geschrieben wird.		
Description	Mit dieser Methode kann die Definition der Anzahl Nachkommastellen des Attributfeldes abgefragt werden.		
Returns	<code>SET_FAILURE</code>	(-14)	Bei einem Fehler.
	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.
Notes	Das SETAttribute-Objekt muss vorher mit <code>Init()</code> Initialisiert worden sein.		

5.16 GetName()

Interface	<code>long GetName(BSTR FAR* value)</code>		
Parameters	<code>value</code> - Platzhalter, in welchen der Name des Attributes geschrieben wird.		
Description	Mit dieser Methode kann der Name des Attributes abgefragt werden.		
Returns	<code>SET_FAILURE</code>	(-14)	Bei einem Fehler.
	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.
Notes	Das SETAttribute-Objekt muss vorher mit <code>Init()</code> Initialisiert worden sein.		

5.17 GetDescription()

Interface	<code>long GetDescription(BSTR FAR* value)</code>		
Parameters	<code>value</code> - Platzhalter, in welchen die Beschreibung des Attributes geschrieben wird.		
Description	Mit dieser Methode kann die Beschreibung des Attributes abgefragt werden.		
Returns	<code>SET_FAILURE</code>	(-14)	Bei einem Fehler.
	<code>SET_NO_ERROR</code>	(0)	Bei keinem Fehler.
Notes	Das SETAttribute-Objekt muss vorher mit <code>Init()</code> Initialisiert worden sein.		

5.18 GetMinMaxFloat()

Interface	<code>long GetMinMaxFloat(double FAR* min, double FAR* max)</code>
Parameters	<code>min</code> - Platzhalter, in welchen der min-Wert geschrieben wird. <code>max</code> - Platzhalter, in welchen der max-Wert geschrieben wird.
Description	Mit dieser Methode können Min/Max-Floatwerte, von Attributspalten abgefragt werden.
Returns	<code>SET_FAILURE</code> (-14) Bei einem Fehler. <code>SET_NO_ERROR</code> (0) Bei keinem Fehler.
Notes	Das SETAttribute-Objekt muss vorher mit <code>Init()</code> Initialisiert worden sein.

5.19 GetMinMaxInteger()

Interface	<code>long GetMinMaxInteger(long FAR* min, long FAR* max)</code>
Parameters	<code>min</code> - Platzhalter, in welchen der min-Wert geschrieben wird. <code>max</code> - Platzhalter, in welchen der max-Wert geschrieben wird.
Description	Mit dieser Funktion können Min/Max-Integerwerte, von Attributspalten abgefragt werden.
Returns	<code>SET_FAILURE</code> (-14) Bei einem Fehler. <code>SET_NO_ERROR</code> (0) Bei keinem Fehler.
Notes	Das SETAttribute-Objekt muss vorher mit <code>Init()</code> Initialisiert worden sein.

5.20 GetMinMaxString()

Interface	<code>long GetMinMaxString(BSTR FAR* min, BSTR FAR* max)</code>
Parameters	<code>min</code> - Platzhalter, in welchen der min-Wert geschrieben wird. <code>max</code> - Platzhalter, in welchen der max-Wert geschrieben wird.
Description	Mit dieser Funktion können Min/Max-Stringwerte, von Attributspalten abgefragt werden.
Returns	<code>SET_FAILURE</code> (-14) Bei einem Fehler. <code>SET_NO_ERROR</code> (0) Bei keinem Fehler.
Notes	Das SETAttribute-Objekt muss vorher mit <code>Init()</code> Initialisiert worden sein.

5.21 GetMultipleMinMaxFloat()

Interface	<code>long GetMultipleMinMaxFloat(???? attr_pos_array, double FAR* min, double FAR* max)</code>
Parameters	<code>attr_pos_array</code> - Liste von Attribut-Positionsindizes. <code>min</code> - Platzhalter, in welchen der min-Wert geschrieben wird. <code>max</code> - Platzhalter, in welchen der max-Wert geschrieben wird.
Description	Mit dieser Funktion können Min/Max-Floatwerte, von mehreren Attributspalten abgefragt werden.
Returns	<code>SET_FAILURE</code> (-14) Bei einem Fehler. <code>SET_NO_ERROR</code> (0) Bei keinem Fehler.
Notes	Das SETAttribute-Objekt muss vorher mit <code>Init()</code> Initialisiert worden sein.

6 Return Codes

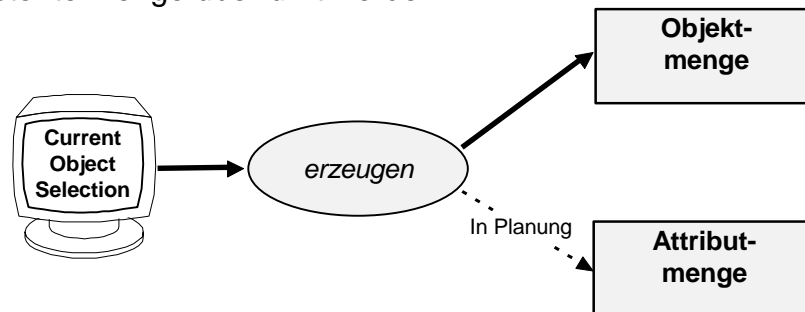
SET_NO_ERROR	0	Die Methode ist ohne Fehler durchgelaufen
SET_EOF	1	End of List
SET_LIST_NOT_INIT	-1	Die Liste ist nicht erzeugt worden
SET_LIST_IS_EMPTY	-2	Die Liste ist leer.
SET_OBJ_NOT_FOUND	-3	Kein Objekt gefunden
SET_POS_IS_WRONG	-4	Der angegebene Positionsindex ist falsch
SET_OBJ_NO_CREATED	-5	Das Objekt ist nicht erzeugt worden
SET_OBJ_IS_NULL	-6	Das Objekt ist NULL
SET_I_OBJ_IS_NULL	-7	Das Interface-Objekt ist NULL
SET_I_OBJ_NOT_INIT	-8	Das Interface-Objekt wurde nicht Initialisiert
SET_NO_DB_CONNECT	-9	Keine Datenbankverbindung
SET_NO_RS_OPEN	-10	Es ist kein RecordSet offen
SET_FIRST_NOT_CALLED	-11	<code>GetFirst()</code> wurde vorher nicht ausgeführt
SET_NOT_CHANGED	-12	Es wurden keine Änderungen gemacht
SET_NO_INSERT	-13	Es wurde kein Datensatz eingefügt
SET_FAILURE	-14	Allgemeiner Fehler

7 Operationen

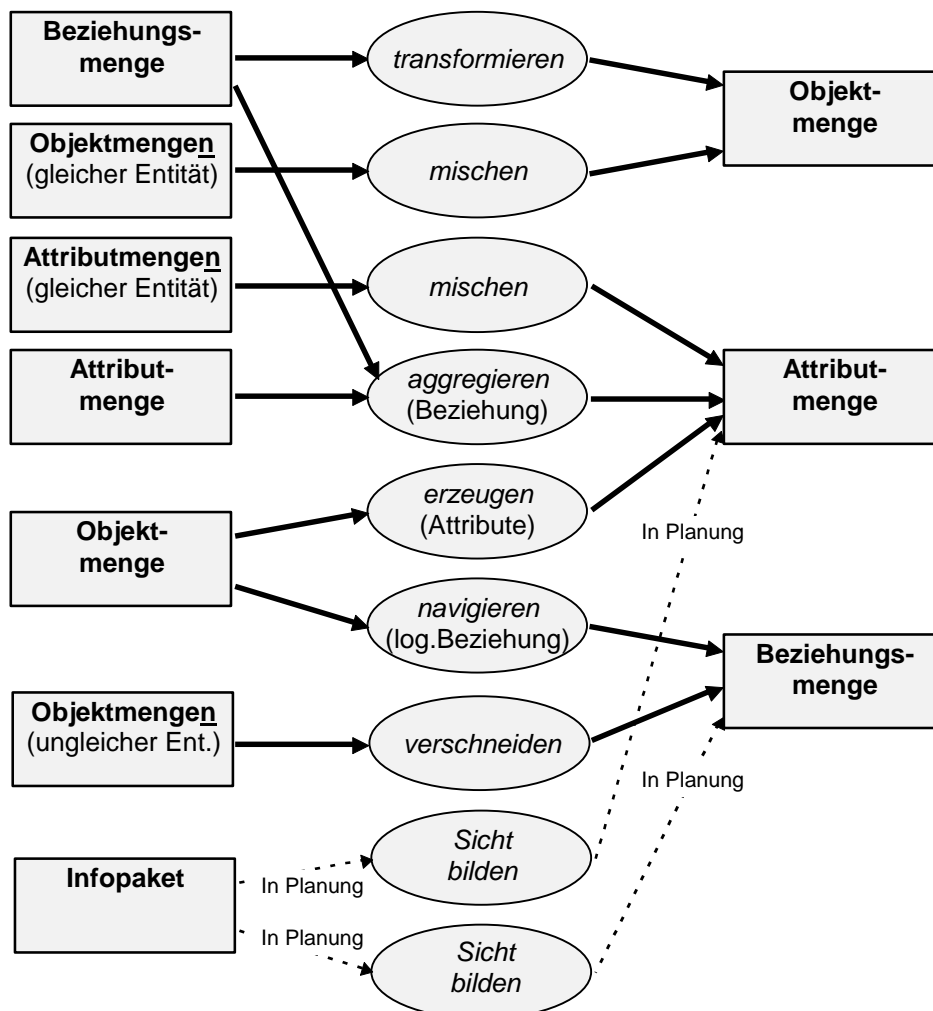
7.1 Operationen mit Mengen

7.1.1 Menge erzeugen

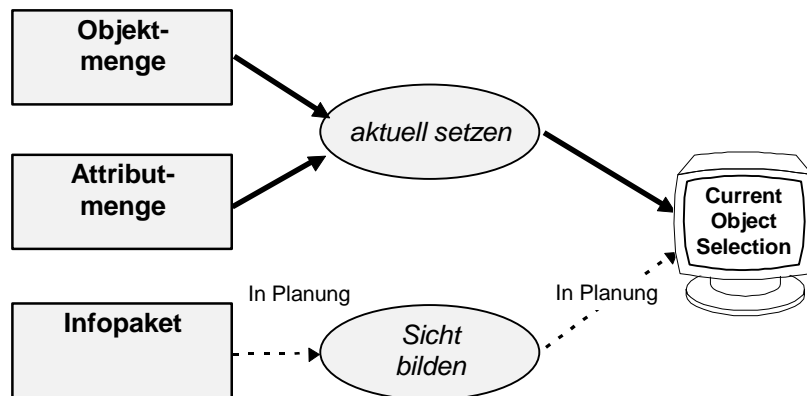
- Die temporäre Menge (Current Object Selection) kann mittels der Operation 'erzeugen' in eine persistente Menge überführt werden.



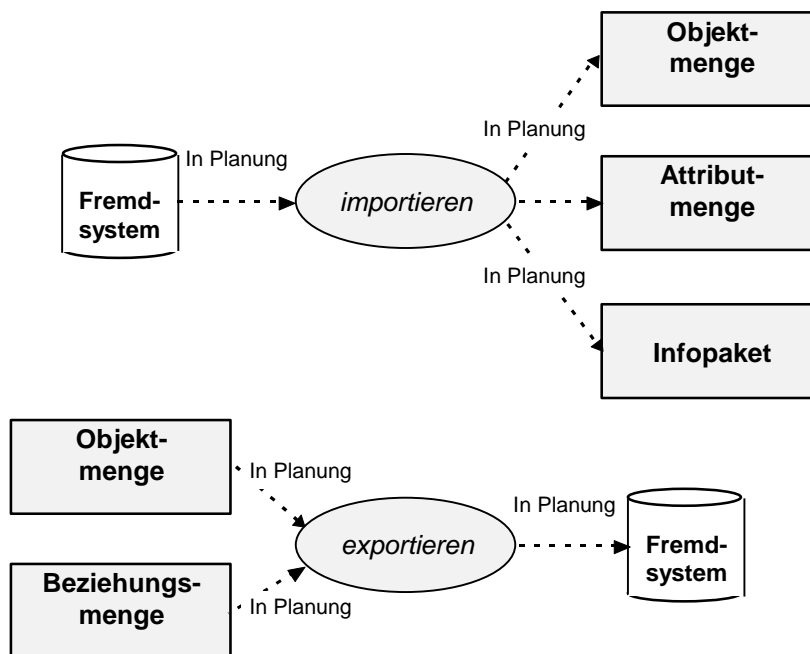
- Sind bereits Mengen im System gespeichert, so können verschiedenste Operation angewendet werden, um diese Mengen zu ändern oder diese als Input zu verwenden, um eine neue Menge zu erzeugen.



- Die Menge selber kann über die Operationen 'aktuell Setzen' und 'Sicht bilden' in eine temporäre Menge (Current Object Selection) überführt werden.



- Eine Menge kann von einem RBS-Fremdsystem 'importiert' oder in dieses 'exportiert' werden.



7.1.2 Menge kopieren

Alle Mengen, auf denen ein lesender Zugriff besteht, können kopiert werden. D.h. der Typ der Inputmenge entspricht dem Typ der Outputmenge. Dabei ändert sich nur der Besitzer und eventuell der Besitzstand der Menge.

7.1.3 Menge löschen

Es können alle Mengen, die dem aktuellen User gehören, gelöscht werden.

7.1.4 Attribute erzeugen

Attribute können von bestehenden Attributmengen oder direkt vom RBS, an eine Objekt- oder Attributmenge kopiert werden. Weiter kann ein neues Attribut, über die bereits bestehenden Attribute einer Menge, mittels einer arithmetischen Operation erzeugt werden.

