

Softwaredokumentation

COM-Schnittstellen für das Kernmodul

AXCDL.DLL

Version 11.0

Sämtliche Softwareprodukte sowie alle Zusatzprogramme und Funktionen sind eingetragene und/oder in Gebrauch befindliche Marken der OPTIMAL SYSTEMS GmbH, Berlin oder einer ihrer Gesellschaften. Sie dürfen nur mit gültigem Lizenzvertrag benutzt werden. Die Software sowie die jeweils zugehörige Dokumentation sind nach deutschem und internationalem Recht urheberrechtlich geschützt. Das illegale Kopieren und Verreiben der Software stellt Diebstahl geistigen Eigentums dar und wird strafrechtlich verfolgt. Alle Rechte vorbehalten, einschließlich der Wiedergabe, Übermittlung, Übersetzung sowie Speicherung mit/auf Medien aller Art. Für vorkonfigurierte Testszenarien oder Demo-Präsentationen gilt: Alle Firmennamen und Personen, die in Beispielen (Screenshots) erscheinen, sind frei erfunden. Eventuelle Ähnlichkeiten mit tatsächlich existierenden Firmen und Personen sind zufällig und unbeabsichtigt.

Copyright 1992 – 2023 by

OPTIMAL SYSTEMS GmbH
Cicerostraße 26
D-10709 Berlin

01.06.2021
Version 11.0

Inhalt

COM-Schnittstelle für das Kernmodul AxCDL	6
Einleitung	6
Zweck	6
Anwendungsbereiche der Komponente	6
Schnittstellen der AxCDL	7
Schnittstellen für den Zugriff auf IndexDaten	7
Schnittstelle für die Verwaltung von Zugriffsrechten	7
Schnittstellen für die Nutzerverwaltung	8
Schnittstellen für den Zugriff auf Mappen	8
Schnittstelle für den Zugriff und die Verwaltung von Relationen	9
Schnittstelle für die Anbindung an den Server	9
Konstanten	9
DrtObjectTypeEnum	9
DrtFieldTypeEnum	10
DrtConditionNodeTypeEnum	10
DrtOperatorEnum	10
DrtImportFlagEnum	11
DrtSaveObjFileFlagsEnum	11
DrtLoadObjFileFlagsEnum	11
DrtRestoreFlagsEnum	11
Hinweise zu den Beispielen	11
Allgemeine Beispiele	12
Hierarchie-Verzeichnis	15
Klassenhierarchie	15
Datenstruktur-Verzeichnis	16
Datenstrukturen	16
Datenstruktur-Dokumentation	17
IComCollection Schnittstellenreferenz	17
Ausführliche Beschreibung	17
IDrtAccess Schnittstellenreferenz	19
Ausführliche Beschreibung	19
Öffentliche Methoden	19
Dokumentation der Elementfunktionen	19
IDrtAccessEntry Schnittstellenreferenz	22
Ausführliche Beschreibung	22
Öffentliche Methoden	22
Dokumentation der Elementfunktionen	22
IDrtAccessList Schnittstellenreferenz	27
Ausführliche Beschreibung	27
Öffentliche Methoden	27
Dokumentation der Elementfunktionen	27
IDrtField Schnittstellenreferenz	30
Ausführliche Beschreibung	30
Öffentliche Methoden	30
Dokumentation der Elementfunktionen	30
IDrtFieldConditionNode Schnittstellenreferenz	35
Ausführliche Beschreibung	35

Öffentliche Methoden	35
Dokumentation der Elementfunktionen	35
IDrtFields Schnittstellenreferenz	40
Ausführliche Beschreibung	40
Öffentliche Methoden	40
Dokumentation der Elementfunktionen	40
IDrtGroup Schnittstellenreferenz	42
Ausführliche Beschreibung	42
Öffentliche Methoden	42
Dokumentation der Elementfunktionen	42
IDrtGroups Schnittstellenreferenz	46
Ausführliche Beschreibung	46
Öffentliche Methoden	46
Dokumentation der Elementfunktionen	46
IDrtMultiField Schnittstellenreferenz	47
Ausführliche Beschreibung	47
Öffentliche Methoden	47
Dokumentation der Elementfunktionen	47
IDrtObject Schnittstellenreferenz	50
Ausführliche Beschreibung	50
Öffentliche Methoden	50
Dokumentation der Elementfunktionen	51
IDrtObjects Schnittstellenreferenz	62
Ausführliche Beschreibung	62
Öffentliche Methoden	62
Dokumentation der Elementfunktionen	62
IDrtObjectTypeConditionNode Schnittstellenreferenz	63
Ausführliche Beschreibung	63
Öffentliche Methoden	63
Dokumentation der Elementfunktionen	63
IDrtPortfolio Schnittstellenreferenz	66
Ausführliche Beschreibung	66
Öffentliche Methoden	66
Dokumentation der Elementfunktionen	66
IDrtPortfolioMgr Schnittstellenreferenz	73
Ausführliche Beschreibung	73
Öffentliche Methoden	73
Dokumentation der Elementfunktionen	73
IDrtPortfolios Schnittstellenreferenz	76
Ausführliche Beschreibung	76
Öffentliche Methoden	76
Dokumentation der Elementfunktionen	76
IDrtQueries Schnittstellenreferenz	77
Ausführliche Beschreibung	77
Öffentliche Methoden	77
Dokumentation der Elementfunktionen	77
IDrtQuery Schnittstellenreferenz	81
Ausführliche Beschreibung	81
Öffentliche Methoden	81
Dokumentation der Elementfunktionen	81
IDrtRelation Schnittstellenreferenz	84
Ausführliche Beschreibung	84
Öffentliche Methoden	84
Dokumentation der Elementfunktionen	84
IDrtRelationMgr Schnittstellenreferenz	94
Ausführliche Beschreibung	94
Öffentliche Methoden	94
Dokumentation der Elementfunktionen	94
IDrtRelations Schnittstellenreferenz	101
Ausführliche Beschreibung	101
Öffentliche Methoden	101

Dokumentation der Elementfunktionen.....	101
IDrtRelationText Schnittstellenreferenz.....	102
Ausführliche Beschreibung.....	102
Öffentliche Methoden.....	102
Dokumentation der Elementfunktionen.....	102
IDrtRelationTexts Schnittstellenreferenz.....	107
Ausführliche Beschreibung.....	107
Öffentliche Methoden.....	107
Dokumentation der Elementfunktionen.....	107
IDrtResult Schnittstellenreferenz.....	108
Ausführliche Beschreibung.....	108
Öffentliche Methoden.....	108
Dokumentation der Elementfunktionen.....	108
IDrtResults Schnittstellenreferenz.....	110
Ausführliche Beschreibung.....	110
IDrtSecurity Schnittstellenreferenz.....	111
Ausführliche Beschreibung.....	111
Öffentliche Methoden.....	111
Dokumentation der Elementfunktionen.....	111
IDrtSession Schnittstellenreferenz.....	114
Ausführliche Beschreibung.....	114
Öffentliche Methoden.....	114
Dokumentation der Elementfunktionen.....	115
IDrtTableField Schnittstellenreferenz.....	128
Ausführliche Beschreibung.....	128
Öffentliche Methoden.....	128
Dokumentation der Elementfunktionen.....	128
IDrtUser Schnittstellenreferenz.....	131
Ausführliche Beschreibung.....	131
Öffentliche Methoden.....	131
Dokumentation der Elementfunktionen.....	132
IDrtUserMgr Schnittstellenreferenz.....	142
Ausführliche Beschreibung.....	142
Öffentliche Methoden.....	142
Dokumentation der Elementfunktionen.....	142
IDrtUsers Schnittstellenreferenz.....	147
Ausführliche Beschreibung.....	147
Öffentliche Methoden.....	147
Dokumentation der Elementfunktionen.....	147

COM-Schnittstelle für das Kernmodul AxCDL

Einleitung

Zweck

Dieses Dokument enthält die Dokumentation der COM-Schnittstellen für das Kernmodul AxCDL. Mithilfe dieser Komponenten ist der Zugriff auf die entsprechenden Funktionalitäten aus z.B. einer Visual Basic, VB-Script oder auch C++ Umgebung möglich. Diese Komponente bietet eine einfache und effiziente Anbindung an die enaio® Server - Namespaces DMS (Document Management System) und ADM (Administration) aus Clientkomponenten und Serverscripts via COM Interface und XML Interface.

DMS - Namespace

Der Namespace wird geliefert von der DMS Engine.

Die Engine bildet die zentrale Instanz zum Handling von:

- Indexdaten von DMS - Objekten (Dokumenten, Register, Ordner)
- Dokumentdateien
- Mappen
- Relationen
- Zugriffsrechten (Security Descriptoren)

ADM - Namespace

Der Namespace wird geliefert vom Server - Kernel. Zu seinen Aufgaben gehört u.a. die Administration von Nutzerprofilen.

Auslesen von Benutzern, Gruppen und entsprechenden Profilen

Anwendungsbereiche der Komponente

in Skript Umgebungen im Client / Server. z.B. Skripte für den enaio® client.

die Initialisierung der Komponente erfolgt direkt von der Skriptengine

als COM Komponente für Anwendungsprogramme

die Initialisierung der Komponente hat in den jeweiligen Programmen zu erfolgen (Beispiel: [IDrtSession.Login](#).)

Schnittstellen der AxCDL

Schnittstellen für den Zugriff auf IndexDaten

Beliebige Objekttypen können recherchiert, erstellt, gelöscht und bearbeitet werden. Der Zugriff auf die Dokumentdateien von Dokumentdatentypen ist möglich.

COM - Objekte für Indexdaten

Objekte zur Anfrage von Indexdaten
Objekte als Antwort von Indexdatenabfragen
Objekte für Bedingungen

Objekte von Indexdaten

[IDrtObjects](#) (*Collection mit IDrtObject - Elementen*)
[IDrtObject](#) (*Datenelement für Objekte*)
[IDrtFields](#) (*Collection mit IDrtObject - Elementen*)
[IDrtField](#) (*Datenelement für ein Objektfeld*)
[IDrtMultiField](#) (*Spezialfeld: Mehrfachfeld*)
[IDrtTableField](#) (*Spezialfeld: Tabellenfeld*)

Objekte zur Anfrage

[IDrtQueries](#) (*Datenelement für ein Anfrageobjekt*)
[IDrtQuery](#) (*Collection mit IDrtQueries - Elementen*)

Objekte für Bedingungen

[IDrtObjectTypeConditionNode](#) (*Bedingung für ein Objekttyp*)
[IDrtFieldConditionNode](#) (*Feldbedingungen*)

Schnelle Verarbeitung einer Serverantwort

[IDrtResults](#) (*Collection mit IDrtResult - Elementen*)
[IDrtResult](#) (Beschreibt ein einzelnes Objekt der Ergebnisliste einer Datenanfrage)

Schnittstelle für die Verwaltung von Zugriffsrechten

COM - Objekte für Security Descriptoren

Manager Objekt
Security Descriptoren

Manager Objekt

[IDrtAccess](#) (Manager zum Verwalten der Zugriffsrechte auf einzelne Objekte: Erstellen, Abfragen, Löschen)

Security Descriptoren

[IDrtAccessList](#) (*Collection mit IDrtAccessEntry - Elementen*)
[IDrtAccessEntry](#) (*Datenelement für Security Descriptor*)

Schnittstellen für die Nutzerverwaltung

Zur Nutzerverwaltung gehören das Anlegen und Löschen von Benutzern und Benutzergruppen, sowie das Recherchieren nach einzelnen Benutzern und Gruppen und ihrer Eigenschaften.

COM Objekte für die Nutzerverwaltung

Manager Objekte
Benutzer Objekte
Gruppen Objekte

Manager Objekt

[IDrtSecurity](#) (Manager zum Abfragen von Nutzer- und Gruppenlisten, sowie Nutzer- und Gruppenattributen)
[IDrtUserMgr](#) (Manager zum Verwalten von Systembenutzern und deren Benutzergruppen)

Benutzer Objekte

[IDrtUsers](#) (*Collection mit IDrtUser - Elementen*)
[IDrtUser](#) (*Datenelement für Benutzereigenschaften*)

Gruppen Objekte

[IDrtGroups](#) (*Collection mit IDrt Group - Elementen*)
[IDrtGroup](#) (*Datenelement für Gruppeneigenschaften*)

Schnittstellen für den Zugriff auf Mappen

Mappen (sogenannte Portfolios) beinhalten eine beliebige Sammlung von Dokumenten, Registern und Ordnern.

COM Objekte für die Mappen

Manager Objekt
Mappen Objekte

Manager Objekt für Mappen

[IDrtPortfolioMgr](#) (Manager zum Verwalten von Mappen (Portfolios): Erstellen, Abfragen, Ändern, Löschen)

Mappen Objekte

[IDrtPortfolios](#) (*Collection mit IDrtPortfolio - Elementen*)
[IDrtPortfolio](#) (*Datenelement für ein Portfolio*)

Schnittstelle für den Zugriff und die Verwaltung von Relationen

Relationen sind spezielle Verknüpfungen zwischen Dokumenten, Registern und Ordnern. Es können Relationen für die Verknüpfung beliebiger Objekttypen eingerichtet werden.

COM Objekte für Relationen

Manager Objekt
Relations Objekte
Relationstyp Objekte

Manager für Relationen

[IDrtRelationMgr](#) (Manager zum Verwalten der Relationsobjekte: Erstellen, Abfragen, Ändern, Löschen)

Relations Objekte

[IDrtRelations](#) (Collection mit IDrtRelation - Elementen)
[IDrtRelation](#) (Datenelement für die Verlinkung zweier Objekte)

Relationstypen Objekte

[IDrtRelationsTexts](#) (Collection mit IDrtRelationText - Elementen)
[IDrtRelationsText](#) (Datenelement für Relationstyp)

Schnittstelle für die Anbindung an den Server

Zentrale Serververbindung

[IDrtSession](#) (Manager für eine Serververbindung, Kommunikation zum Server, Erzeugen anderer Managerobjekte)

Konstanten

Für verschiedene Funktionen dieser Komponente sind unterschiedliche Konstanten als Rückgabewerte oder zur genaueren Konfiguration dieser Methode definiert.

[DrtObjectTypeEnum](#) (Ist ein Objekt ein Ordner, Register oder Dokument.)
[DrtFieldTypeEnum](#) (Ist ein Feld ein Mehrfachfeld oder Tabellenfeld.)
[DrtConditionNodeTypeEnum](#) (Konstanten für Bedingungen.)
[DrtOperatorEnum](#) (Konstanten für die Vergleichsoperatoren für Feldbedingungen)
[DrtImportFlagEnum](#) (Option für das Einfügen, Ändern und Löschen von Objekten.)
[DrtSaveObjFileFlagsEnum](#) (Option beim Speichern von Dokumentdateien.)
[DrtLoadObjFileFlagsEnum](#) (Option beim Laden von Dokumentdateien.)
[DrtRestoreFlagsEnum](#) (Option beim Wiederherstellen gelöschter Objekte.)

DrtObjectTypeEnum

Gibt an, ob ein [IDrtObject](#) ein Ordner, Register oder ein Dokument ist.

Diese Werte werden der Eigenschaft `Type` eines [IDrtObject](#) gehalten.

`DrtObjectType_Folder = 0 (Ordner)`
`DrtObjectType_Register = 1 (Register)`
`DrtObjectType_Document = 2 (Dokument)`

DrtFieldTypeEnum

Gibt an, ob ein [IDrtField](#) ein einfaches Feld, ein Mehrfachfeld oder ein Tabellenfeld ist.

Diese Werte werden der Eigenschaft `Type` eines [IDrtField](#) gehalten. Bei Mehrfach- oder Tabellenfeldern kann dann über die Attribute `Multifield` oder `TableField` auf die speziellen Eigenschaften eines entsprechenden Feldes zugegriffen werden.

`DrtFieldType_SimpleField = 0 (Einfaches Feld)`
`DrtFieldType_MultiField = 1 (Mehrfachfeld)`
`DrtFieldType_TableField = 2 (Tabellenfeld)`

DrtConditionNodeTypeEnum

Gibt den Typ eines Bedingungsknotens an.

Für Bedingungen auf einen Objekttyp ([IDrtObjectTypeConditionNode](#)) sind nur die Werte `DrtConditionNodeType_Leaf` und `DrtConditionNodeType_Flat` verfügbar. Eine Bedingung vom Typ `DrtConditionNodeType_Flat` enthält weitere Bedingungen vom Typ `DrtConditionNodeType_Leaf`. Eine Bedingung vom Typ `DrtConditionNodeType_Leaf` ist durch einen festgelegten Objekttyp und die Angabe eine Feldbedingung definiert. Die einzelnen Bedingungen vom Typ `DrtConditionNodeType_Leaf` sind durch ODER verknüpft wenn sie sich auf denselben Objekttyp beziehen, oder durch UND verknüpft, wenn sie sich auf verschiedene Objekttypen beziehen.

Für Bedingungen auf ein Indexdatenfeld [IDrtFieldConditionNode](#) sind nur die Werte `DrtConditionNodeType_Leaf`, `DrtConditionNodeType_Or` und `DrtConditionNodeType_And` verfügbar. Die Bedingung vom Typ `DrtConditionNodeType_Leaf` ist die erste angegebene Feldbedingung, folgen weitere Bedingungen, dann ist über die beiden anderen Typen festgelegt, wie diese Bedingungen mit den Vorgängerbedingungen verknüpft werden sollen.

`DrtConditionNodeType_Leaf = 0` (keine weiteren Bedingungen vorhanden)
`DrtConditionNodeType_Or = 1` (folgende Bedingung ODER verknüpft)
`DrtConditionNodeType_And = 2` (folgende Bedingung UND verknüpft)
`DrtConditionNodeType_Flat = 2` (aktuelles Bedingungelement ist nur eine Liste weitere Bedingungen)

DrtOperatorEnum

Werte für Operatoren von Feldbedingungen ([IDrtFieldConditionNode](#))

Diese werden verwendet, um Bedingungen für Suchanfragen zu stellen. Siehe auch Beispiel: Indexdatenabfrage ([IDrtSession](#)).

`DrtOperatorEnum_Equ = 0` (Gleichheitsoperator: $A = B$)
`DrtOperatorEnum_Neq = 1` (Ungleich - Operator: $A \neq B$)
`DrtOperatorEnum_Les = 2` (Kleiner als - Operator: $A < B$)
`DrtOperatorEnum_Leq = 3` (Kleiner/Gleich - Operator: $A \leq B$)
`DrtOperatorEnum_Gre = 4` (Größer als - Operator: $A > B$)
`DrtOperatorEnum_Geq = 5` (Größer/Gleich - Operator: $A \geq B$)
`DrtOperatorEnum_Bet = 6` (Operator: $A \text{ BETWEEN } B, C$)
`DrtOperatorEnum_Nbe = 7` (Operator: $A \text{ NOT BETWEEN } B, C$)
`DrtOperatorEnum_In = 8` (Enthalten in - Operator: $A \text{ IN } [B, C, D]$)
`DrtOperatorEnum_Nin = 9` (Nicht enthalten in - Operator: $A \text{ IN } [B, C, D]$)

DrtImportFlagEnum

Optionen für die Funktionen *Insert*, *Update*, *Delete* und *Move* von [IDrtSession](#).

DrtImportFlag_ReplaceFiles = 0x00000080 (Nur Update: Vorhandene Dateien des DrtObject werden ersetzt durch die Dateien in der Dateiliste, andernfalls werden die Dateien aus der Dateiliste zu den vorhandenen Dateien ergänzt.)
 DrtImportFlag_UpdateAllFields = 0x00008000 (Nicht initialisierte Felder werden leer überschrieben)
 DrtImportFlag_ReplaceMultiFields = 0x00020000 (*Ersetze Mehrfachfelder*)
 DrtImportFlag_ReplaceTableFields = 0x00040000 (*Ersetze Tabellenfelder*)
 DrtImportFlag_DeleteCascading = 0x00100000 (Objekt soll einschließlich aller seiner Unterobjekte gelöscht werden)
 DrtImportFlag_DontCheckExistence = 0x00200000 (Existenztest entfällt (in Kombination mit DrtImportFlag_HardDelete))
 DrtImportFlag_HardDelete = 0x00400000 (Objekt soll physikalisch gelöscht werden)
 DrtImportFlag_Encrypt = 0x00800000 (Die Dateien werden verschlüsselt zum Server übertragen)

DrtSaveObjFileFlagsEnum

Option beim Speichern eines Dokumentdatei.

Je nach Option können neue Dateien entweder an die vorhandenen Dateien des Dokumentes angehängt werden, oder die vorher vorhandenen Dateien des Dokumentes überschrieben werden. (vgl. [IDrtSession](#).)

DrtSaveObjFile_Standard = 0x00000000 (*Überschreibe vorhandene Dateien*)
 DrtSaveObjFile_Append = 0x00000001 (Füge Dateien an vorhandene Dateien an)
 DrtSaveObjFile_Encrypt = 0x40000000 (Die Dateien werden verschlüsselt zum Server übertragen)
 DrtSaveObjFile_Checkin = 0x80000000 (*Dokument wird eingecheckt*)

DrtLoadObjFileFlagsEnum

Option beim Laden einer Dokumentdateien.

Je nach Option können entweder nur Dokumentdateien oder ebenfalls auch Dia (Vorschau-) Dateien für das aktuelle Dokument angefragt werden. Diese Dateien können auf dem Server zum Bearbeiten markiert werden. (vgl. [IDrtSession](#).)

DrtLoadObjFile_Both = 0x00000000 (sowohl Dias als auch Dokumentdateien)
 DrtLoadObjFile_Doc = 0x00000001 (*nur Dokumentdateien*)
 DrtLoadObjFile_Dia = 0x00000002 (*nur Dia-Dateien*)
 DrtLoadObjFile_NoDownload = 0x40000000 (*Datei wird nicht erneut übertragen*)
 DrtLoadObjFile_CheckOut = 0x80000000 (Dokument wird auf dem Server zum Bearbeiten markiert)

DrtRestoreFlagsEnum

Option beim Wiederherstellen von gelöschten Objekten aus dem Papierkorb.

Es kann je nach Option entweder nur das aktuell angegebene Objekt wiederhergestellt werden, oder wenn dieses ein Ordner oder Register ist, kann dieses Objekt einschliesslich der in ihm vorhandenen Unterobjekte wiedergestellt werden. (vgl. [IDrtSession](#).)

DrtRestore_Single = 0 (nur das angefragte Objekt)
 DrtRestore_All = 1 (angefragtes Objekt einschliesslich aller Unterobjekte)

Hinweise zu den Beispielen

Alle Beispiele, die in dieser Dokumentation aufgeführt werden, wurden in der Scriptsprache Microsoft VBScript geschrieben.

Sollte man diese Beispiele verwenden, so muss man darauf achten, dass das DRTSession Objekt für alle Skriptfunktionen die zentrale Kommunikationsinstanz zum Server darstellt ([IDrtSession](#)). In Client und Serverskripten wird dieses DrtSession - Objekt bereits voninitialisiert zur Verfügung gestellt. Zudem sollte der komplette Code in einer Visual Basic Script Datei sein, um das Objekt bis zum Ende benutzen zu können. Die Datei muss im enaio-Verzeichnis \Client32\ liegen. Sollte man das Objekt löschen mit beispielsweise (Set DrtSession = nothing) und dann bei der nächsten Methode wieder erneut erstellen so funktioniert der Beispielcode nicht mehr.

Allgemeine Beispiele

Login / Logout (->IDrtSession)
 Indexdatenabfrage (->IDrtSession)
 Mappen (->IDrtPortfolioMgr)
 Relationen (->IDrtRelationMgr)
 Benutzerrechte (->IDrtSecurity, ->IDrtUserMgr)

Login / Logout (->IDrtSession)

```
Dim DrtSession           //Objekt was erstellt werden muss

Set DrtSession = CreateObject("axcdl.DrtSession")

bstrUserName = "Mustermann"      //Benutzername
bstrPassword = "muster"          //Password
bstrServerName = "Server1"       //Servername
bstrServerPort = "4000"          //Serverport
bstrAppName = "Login"            //Name der Applikation
bstrLicence = "ABC"              //Licence Name
hWnd = 0                          //Handle für das Fenster
lDbCursorCount = 1               //Datenbankzeiger, Standard auf 1

DrtSession.Login bstrUserName, bstrPassword, bstrServerName, bstrServerPort, bstrAppName,
bstrLicence, hWnd, lDbCursorCount
```

Indexdatenabfrage (->IDrtSession)

```
Dim Queries
Dim Query1, Query2
Dim Objects, RObjects
Dim Object

Set Queries = CreateObject("AxCdL.DrtQueries") //Objekt Anfrage
Set Query1 = CreateObject("AxCdL.DrtQuery")    //Objekt für die 1te Anfrage
Set Query2 = CreateObject("AxCdL.DrtQuery")    //Objekt für die 2te Anfrage
Set Objects = CreateObject("AxCdL.DrtObjects") //Objekt für die Anfrageinformationen
Set RObjects = CreateObject("AxCdL.DrtObjects") //Objekt für das Ergebnis der Anfrage

Set Obj1 = CreateObject("AxCdL.DrtObject")     //Angefragter Objekttyp 1
Set Obj2 = CreateObject("AxCdL.DrtObject")     //Angefragter Objekttyp 2
Set Field1 = CreateObject("AxCdL.DrtField")    //Objekt für die Speicherung der
Feldinformationen
Set Field2 = CreateObject("AxCdL.DrtField")
Set Field3 = CreateObject("AxCdL.DrtField")

Set NodeCond1 = CreateObject("AxCdL.DrtObjectTypeConditionNode") //Bedingung für Objekttyp
Set FieldCondsOr = CreateObject("AxCdL.DrtFieldConditionNode") //Liste mehrerer
Feldbedingungen
Set FieldCond1 = CreateObject("AxCdL.DrtFieldConditionNode") //Bedingung für Indexdatenfeld
Set FieldCond2 = CreateObject("AxCdL.DrtFieldConditionNode") //Bedingung für Indexdatenfeld

Field1.Name = "Name"                                //Festlegen der angefragten
Indexdatenfelder
Field2.Name = "Datum"
Field3.Name = "Beschreibung"

With Obj1
    .Name = "Farbbild"                                //Festlegung des Objekttyps über Name des
Objekttyps
```

```

        .CabinetId = 4                                //Objekttyp ist im Schrank mit Objekttyp 4
definiert
        .Fields.Add 0,Field1                          //Werte zu Indexdatenfeld "Name" werden
angefragt
        .Fields.Add 1,Field2                          //Werte zu Indexdatenfeld "Datum" werden
angefragt
    End With

    With Obj2
        .ObjectTypeID = 131073                        //SW-Bild
        .ID=490                                       //Festlegung des Objekttyps über ID
    End With                                       //als Suchbedingung wird hier direkt eine
bekannte ID angegeben
    End With                                       //Hier werden alle Felder angefragt

    With FieldCond1
        .NodeType = 0
        .Field = Field1
    EQUAL "Test"
        .Operator = 0
        .AddValue "Test"
    End With                                       //Initialisieren einer Feldbedingungen
                                                //
                                                //Als Bedingung wird verwendet: "Name"

    With FieldCond2
        .NodeType = 0
        .Field = Field3
    "Beschreibung" NOT EQUAL "Test"
        .Operator = 1
        .AddValue "Test"
    End With                                       //Initialisieren einer Feldbedingungen
                                                //
                                                //Als Bedingung wird verwendet:

    With FieldCondsOr
    (cond1 OR cond2)
        .NodeType = 1
        .Add 0,FieldCond1
        .Add 1,FieldCond2
    End With                                       // Veroderung beider Feldbedingungen

    With NodeCond1
    Anfrage
        .NodeType = 0
        .ObjectType = Obj1
    gesetzt
        .FieldConditions = FieldCondsOr
    End With                                       //Initialisieren der Suchbedingung für die
                                                //nur einzelne Bedingung wird gesetzt
                                                //Bedingung wird für Obj1 "Farbbild"

    Query1.Object = Obj1
    Anfrage
        Query1.Conditions = NodeCond1
    aktuellen Query
        Query2.Object = Obj2
    Anfrage
        Query2.ChildObjectsDepth=2
    Kindobjekte mit ausgegeben werden
        Queries.Add 0,Query1
    und Reihenfolge festlegen
        Queries.Add 1,Query2
                                                //Zuweisen des Anfragetyps zur ersten
                                                //Zuweisen einer Suchbedingung zur
                                                //Zuweisen des Anfragetyps zur zweiten
                                                //Bei der Registeranfrage sollen
                                                //Die Anfragen dem Anfrageobjekt zuweisen

    Set RObjects = DrtSession.GetResultObjects(Queries) //Anfrage ausführen und Ergebnis im
RObjects Objekt speichern

    //.... weitere Verarbeitung der Anfrage mit Ihrem Quelltext

```

Mappen (->DrtPortfolioMgr)

```

Dim MapMgr
Dim maps
Set MapMgr = DrtSession.CreatePortfolioMgr           //Erstelle Instanz des Managers
Set maps = MapMgr.RetrievePortfolios(-1,"", "",0,0,"ROOT") //Ermittle alle Mappen die
"ROOT" erzeugt hat

```

Relationen (->IDrtRelationMgr)

```

Dim relations
Dim text
Dim message
Dim RelMgr
Set RelMgr = DrtSession.CreateRelationMgr           //Erstelle Instanz des Managers
Dim mm
Set mm=createobject("oxvbbas.functions")           //Hilfsfunktion zum Handling von
Timestamps

Set relations = RelMgr.RetrieveRelations(-1,123,-1,444,1,0,0,-1,-1,"",-1,-1,0,0,0,0)//
Suche alle Relationen zwischen Objekttyp 123 und 444
For i=0 To relations.count-1
    Set rel = relations(i)                         //Ermittle Treffer
    GetRelText rel.Obj_typ2,rel.Obj_typ2,rel.RelTextID //Ermittle Name des Relationstyps
    message = message + vbTab + mm.ConvertTimestamp(rel.Created) + vbTab +
mm.ConvertTimestamp(rel.Modifytime)                //Ausgabestring zusammenbauen
    text = text & "to Object(" & rel.Obj_id2 & "," & rel.Obj_typ2 & "):" & message & vbCrLf
Next
If relations.count=0 Then                          //Ausgabe des entsprechenden
Ergebnisstrings
    MsgBox "keine Relationen gefunden"
Else
    MsgBox text
End if

Sub GetRelText(ot1, ot2, id)
Set reltexts = RelMgr.RetrieveRelTexts(ot1, ot2, 1, 0)
For j=0 To reltexts.Count-1
If reltexts(j).RelTextID=id Then
    message = reltexts(j).TextTo                    //Ermittle Namen des Relationstyps
End If
Next
End Sub

```

Benutzerrechte (->IDrtSecurity, ->IDrtUserMgr)

```

Dim UserMgr
Dim users
Dim user
Dim groups
Set UserMgr = DrtSession.CreateSecurityObj           // Erzeuge Manager
Set users = UserMgr.GetUserList                     // Hole Liste aller Benutzer im System
For i=0 To users.count-1
    Set user = users(i)
    Set groups = UserMgr.GetGroupsOfUser(user)        // Hole Benutzergruppen zum aktuellen
Benutzer

    // .... weitere Verarbeitung dieser Benutzergruppen
Next

```

Hierarchie-Verzeichnis

Klassenhierarchie

Die Liste der Ableitungen ist -mit Einschränkungen- alphabetisch sortiert:

IComCollection	17
IDrtAccess	19
IDrtAccessEntry	22
IDrtAccessList	27
IDrtField	30
IDrtFieldConditionNode.....	35
IDrtFields	40
IDrtGroup	42
IDrtGroups	46
IDrtMultiField	47
IDrtObject	50
IDrtObjects	62
IDrtObjectTypeConditionNode.....	63
IDrtPortfolio	66
IDrtPortfolioMgr	73
IDrtPortfolios	76
IDrtQueries	77
IDrtQuery	81
IDrtRelation	84
IDrtRelationMgr	94
IDrtRelations	101
IDrtRelationText	102
IDrtRelationTexts	107
IDrtResult	108
IDrtResults	110
IDrtSecurity	111
IDrtSession	114
IDrtTableField	128
IDrtUser	131
IDrtUserMgr	142
IDrtUsers	147

Datenstruktur-Verzeichnis

Datenstrukturen

Hier folgt die Aufzählung aller Datenstrukturen mit einer Kurzbeschreibung:

IComCollection (Basisschnittstelle für alle Listen)	17
IDrtAccess (Beschreibt die Objektliste für den Sicherheitszugriff)	19
IDrtAccessEntry (Definiert den Sicherheitsflag und die Benutzer/Gruppen GUID's)	22
IDrtAccessList (Zusammenstellung von IDrtAccessEntry Objekte)	27
IDrtField (Beschreibt ein einzelnes Feldobjekt)	30
IDrtFieldConditionNode (Beschreibt Methoden zur Parametrisierung von Anfragen)	35
IDrtFields (Zusammenstellung von Objekten vom Typ IDrtField)	40
IDrtGroup (Darstellung des Gruppen Objektes)	42
IDrtGroups (Darstellung der Gruppenlisten)	46
IDrtMultiField (Beschreibt ein spezielles Feldobjekt: Mehrfachfeld)	47
IDrtObject (Beschreibt ein einzelnes Indexdatenobjekt)	50
IDrtObjects (Zusammenstellung von Objekten vom Typ IDrtObject)	62
IDrtObjectTypeConditionNode (Beschreibt Methoden zur Parametrisierung von Anfragen)	63
IDrtPortfolio (Darstellung des Portfolio Objekts)	66
IDrtPortfolioMgr (Beschreibt Methoden zur Bearbeitung von Mappen)	73
IDrtPortfolios (Zusammenstellung von IDrtPortfolio Objekte)	76
IDrtQueries (Zusammenstellung von Objekten vom Typ IDrtQuery)	77
IDrtQuery (Objekt für Einschränkung von Recherchen)	81
IDrtRelation (Darstellung von Relations Objekten)	84
IDrtRelationMgr (Beschreibt Methoden zur Bearbeitung von Relationen und Relationstext - Objekten)	94
IDrtRelations (Zusammenstellung von IDrtRelation Objekt)	101
IDrtRelationText (Darstellung von RelationText Objekten)	102
IDrtRelationTexts (Zusammenstellung von IDrtRelationText Objekte)	107
IDrtResult (Beschreibt ein einzelnes Objekt der Ergebnisliste einer Datenanfrage)	108
IDrtResults (Zusammenstellung von Objekten vom Typ IDrtResult)	110
IDrtSecurity (Beschreibt die variierenden Informationen über Benutzer und Benutzergruppen)	111
IDrtSession (Verwaltet die Verbindung zum Server)	114
IDrtTableField (Beschreibt ein spezielles Feldobjekt: Tabellenfeld)	128
IDrtUser (Darstellung von Benutzerobjekten)	131
IDrtUserMgr (Einrichtung der Benutzerprofile)	142
IDrtUsers (Darstellung von Benutzerlisten)	147

Datenstruktur-Dokumentation

IComCollection Schnittstellenreferenz

Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Liste von Objekten. Dies ist die Basisschnittstelle für alle Listen. Die Schnittstelle stellt folgende Methoden bereit:

HRESULT count ([out, retval] long *pVal)

Beschreibung: Diese Methode liefert die Anzahl der Listenelemente.

Rückgabewerte:

LONG pVal

Beispiel:

```
CountElements = MyList.count
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Item ([in]VARIANT varItem , [out, retval] T** pVal)

Beschreibung: Diese Methode liefert ein Listenelement anhand des Listenindex (Position) oder des eindeutigen Schlüssels.

Parameter:

VARIANT varItem

Rückgabewerte:

T* pVal

```
Set Element = MyList.item(index)
```

Siehe: [Allgemeine Beispiele](#)

Die Methoden **add** und **remove** können bei statischen Listen nicht verwendet werden.

HRESULT Add ([in]VARIANT varItem , [in] T* pNewVal)

Beschreibung: Mit Hilfe dieser Methode kann ein Objekt der Liste hinzugefügt werden.

Parameter:

VARIANT varItem

Rückgabewerte:

T* pNewVal

```
MyList.add Object.Id, Object
```

Siehe: [Allgemeine Beispiele](#)

HRESULT remove ([in]VARIANT varItem)

Beschreibung: Mit Hilfe dieser Methode kann ein Objekt der aus Liste gelöscht werden. Das Objekt wird über den Listenindex oder den eindeutigen Schlüssel identifiziert.

Parameter:

VARIANT varItem

```
MyList.remove (index)
```

Siehe: [Allgemeine Beispiele](#)

IDrtAccess Schnittstellenreferenz

Abgeleitet von IDispatch.

Ausführliche Beschreibung

Die Zugriffsrechte eines einzelnen Datenobjektes werden von den sogenannten Sicherheitsdescriptoren beschrieben. Damit ist festgelegt welcher Benutzer/Gruppe Zugriff auf das Objekt und Rechte zu einzelnen Aktionen wie Anlegen, Löschen, etc hat. Diese Schnittstelle erlaubt den Zugriff auf diese Sicherheitsdescriptoren.

Öffentliche Methoden

HRESULT [Create](#) ([in] [IDrtAccessList](#) *pACL,[out, retval] long *pResult)
 HRESULT [Set](#) ([in] [IDrtAccessList](#) *pACL,[out, retval] long *pResult)
 HRESULT [Copy](#) ([in] BSTR Destination,[in] long SrcObjectID,[in] long SrcObjectType,[out, retval] long *pResult)
 HRESULT [Read](#) ([in] long Flags,[in] long ObjectID,[in] long ObjectType,[in] BSTR OSUID,[out, retval] [IDrtAccessList](#) **ppACL)
 HRESULT [Delete](#) ([in] long Flags,[in] long ObjectID,[in] long ObjectType,[in] BSTR OSUID,[in] BSTR OSSD,[out, retval] long *pResult)

Dokumentation der Elementfunktionen

HRESULT Create ([in] [IDrtAccessList](#) * pACL, [out, retval] long * pResult)

Beschreibung: Diese Methode erstellt einen neuen Sicherheitsdescriptor für ein DMS Objekt. In [IDrtAccessList](#) ist dabei festgelegt, um welche Objekt (ID und Objekttyp) es sich dabei handelt, sowie auf welche Werte die Sicherheitsdescriptoren festgelegt werden.

Parameter:

*IDrtAccessList** pACL: Zustand von IDrtAccesslist mit gefüllten ObjectType und ObjectID Eigenschaften

Rückgabewerte:

*LONG** pResult

Hinweis: mögliche Fehlercodes

0 - keine Fehler, oder dms.CreateSD job Fehlerrückgabe.

Beispiel:

```
Dim pACL
Set pACL = CreateObject("axcdl.DrtAccessList")
pResult = IDrtAccess.Create(pACL)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Set ([in] [IDrtAccessList](#) * pACL, [out, retval] long * pResult)

Beschreibung: Diese Methode setzt die Zugriffsliste von Objekten. Diese Methode verwendet den Serverjob: dms.SetSD.

Parameter:

*IDrtAccessList** pACL: Zustand von der gefüllten Zusammenstellung der IDrtAccesslist

Rückgabewerte:

*LONG** pResult

Hinweis: mögliche Fehlercodes

0 - keine Fehler, oder dms.SetSD job Fehlerrückgabe.

Beispiel:

```
Dim pACL

Set pACL = CreateObject("axcdl.DrtAccessList")

pResult = IDrtAccess.Set(pACL)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Copy ([in] BSTR *Destination*, [in] long *SrcObjectID*, [in] long *SrcObjectType*, [out, retval] long * *pResult*)

Beschreibung: Diese Methode kopiert die Zugriffslisten von Objekten mit ID und Type zuverlässig zu einer anderen Objektbeschreibung in den Zielstring. Diese Methode verwendet den Serverjob: dms.CopySD.

Parameter:

BSTR Destination :Zielstring beschreibt die Zielliste. Das Format des Strings ist gleich der Ausgabe von IDrtAccesslist::Serialize mit ObjectID und ObjectType Feldern gesetzt. Die Liste von Mehrfachzuständen beziehen sich auf Verkettungen von mehreren Suchstrings.

*LONG*SrcObjectID: source object id

*LONG*SrcObjectType: source object type

Rückgabewerte:

*LONG** pResult

Hinweis:

pResult 0 - keine Fehler, oder dms.CopySD job Fehlerrückgabe.

Beispiel:

```
Destination = ACL1.Serialize()+ACL2.Serialize()+ACL3.Serialize()
```

Beispiel:

```
pResult = IDrtAccess.Copy(Destination, SrcObjectID, SrcObjectType)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Read ([in] long *Flags*, [in] long *ObjectID*, [in] long *ObjectType*, [in] BSTR *OSUID*, [out, retval] [IDrtAccessList](#) ** *ppACL*)

Beschreibung: Die Methode liest und ersetzt die Eigenschaften von spezifischen DrtAccess Objekten gemäß der spezifischen Parameter. Diese Methode verwendet den Serverjob: dms.ReadSD.

Parameter:

*LONG*Flags

0 - zurückgeben ACL von Objekten von Objekt ID und Typ.

1 - zurückgeben ACE (nur 1 Eintrag in der Liste) von osobjaccess von ossd und osuid guids.

*LONG*ObjectID Objekt ID

*LONG*ObjectType Objekt Typ

BSTR OSUID Benutzer guid (benutzt nur Flags == 1)

Rückgabewerte:

*IDrtAccessList*** ppACL: erstellt eine Zugriffsliste vom Objektzustand

Beispiel:

```
ppACL = IDrtAccess.Read(Flags, ObjectID, ObjectType, OSUID)
```

HRESULT Delete ([in] long *Flags*, [in] long *ObjectID*, [in] long *ObjectType*, [in] BSTR *OSUID*, [in] BSTR *OSSD*, [out, retval] long **pResult*)

Beschreibung: Diese Methode löscht die ACE und / oder ACL von osobjaccess DB Tabelle und etwaige Sicherheitsdescriptor GUIDs von der Objekttabelle. Diese Methode verwendet den Serverjob: dms.DeleteSD.

Parameter:

*LONG*Flags

0 - lösche ACL vom Objekt mit der Objekt ID und Type.

In diesem Fall werden beide, osobjaccess und object Tabellen ausgewechselt. In der Objekttabelle wird entsprechend das ossd Feld gesäubert.

1 - lösche ACE von osobjaccess vom ossd und osuid guids. In diesem Fall wechselt nur osobjaccess, ossd guid in der Objekttabelle nach links. Also diese Situation ist gleich: "Objekt hat sd nicht gesetzt".

2 löschen von ACE von object id, type und osuid guids. In diesem Fall werden beide osobjaccess und object Tabellen verändert. In der object Tabelle wird entsprechend das ossd Feld gesäubert. Dateien werden nicht verwendet.

Parameter:

*LONG*ObjectID Objekt ID (benutzt Flags==0,2)

*LONG*ObjectType Objekt Typ (benutzt Flags==0,2)

BSTR OSUID Benutzer oder Gruppen GUID (benutzt Flags==1,2)

BSTR OSSD Sicherheitsbeschreiber GUID (benutzt Flags==1)

Rückgabewerte:

*LONG** pResult

Hinweis: mögliche Fehlercodes

0 - keine Fehler, oder dms.DeleteSD job Fehlerrückgabe.

Beispiel:

```
pResult = IDrtAccess.Delete(Flags, ObjectID, ObjectType, OSUID, OSSD)
```

IDrtAccessEntry Schnittstellenreferenz

Abgeleitet von IDispatch.

Ausführliche Beschreibung

Diese Schnittstelle definiert einen Zugriffssteuerungseintrag, d.h. die einzelnen Zugriffsrechte für einen Benutzer(/Benutzergruppe) und ein DMS Objekt. Es kann hiermit geregelt werden, dass Benutzer Objekte löschen, exportieren oder ändern dürfen.

Siehe auch: [IDrtAccessList](#)

Öffentliche Methoden

HRESULT [Deserialize](#) ([in] BSTR bsXml)
 HRESULT [Serialize](#) ([out, retval] BSTR *pbsXml)
 HRESULT [OSUID](#) ([out, retval] BSTR *pVal)
 HRESULT [OSUID](#) ([in] BSTR newVal)
 HRESULT [IsGroup](#) ([out, retval] VARIANT_BOOL *pVal)
 HRESULT [IsGroup](#) ([in] VARIANT_BOOL newVal)
 HRESULT [ModifyIndex](#) ([out, retval] long *pVal)
 HRESULT [ModifyIndex](#) ([in] long newVal)
 HRESULT [ModifyObject](#) ([out, retval] long *pVal)
 HRESULT [ModifyObject](#) ([in] long newVal)
 HRESULT [ExportObject](#) ([out, retval] long *pVal)
 HRESULT [ExportObject](#) ([in] long newVal)
 HRESULT [DeleteObject](#) ([out, retval] long *pVal)
 HRESULT [DeleteObject](#) ([in] long newVal)
 HRESULT [ModifySecurity](#) ([out, retval] long *pVal)
 HRESULT [ModifySecurity](#) ([in] long newVal)

Dokumentation der Elementfunktionen

HRESULT Deserialize ([in] BSTR *bsXml*)

Beschreibung: Diese Methode initialisiert ein [IDrtAccessEntry](#) - Objekt einschliesslich seiner Attribute und Unterobjekte aus einem XML String.

Parameter:

BSTR bsXml: XML - Input string

Beispiel:

```
IDrtAccessEntry.Deserialize(bsXml)
```

Siehe: [Allgemeine Beispiele](#) , DMSAccess

HRESULT Serialize ([out, retval] BSTR **pbsXml*)

Beschreibung: Diese Methode serialisiert das [IDrtAccessEntry](#) - Objekt einschliesslich seiner Attribute und Unterobjekte vollständig in einen XML String.

Rückgabewerte:

*BSTR**pbsXml: output xml-string

Beispiel:

```
pbsXml = IDrtAccessEntry.Serialize
```

Siehe: [Allgemeine Beispiele](#)

HRESULT OSUID ([out, retval] BSTR * *pVal*)

Beschreibung: Diese Methode erhält die UID des Benutzers oder der Benutzergruppe auf die sich das [IDrtAccessEntry](#) Objekt bezieht.

Rückgabewerte:

*BSTR***pVal*: UID des Benutzers

Beispiel:

```
pVal = IDrtAccessEntry.OSUID
```

Siehe: [Allgemeine Beispiele](#)

HRESULT OSUID ([in] BSTR *newVal*)

Beschreibung: Diese Methode setzt die UID des Benutzers oder der Benutzergruppe auf die sich das [IDrtAccessEntry](#) Objekt bezieht.

Rückgabewerte:

BSTR *pVal*: UID des Benutzers

Beispiel:

```
IDrtAccessEntry.OSUID(pVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT IsGroup ([out, retval] VARIANT_BOOL * *pVal*)

Beschreibung: Diese Methode gibt an, ob es sich bei dem Wert OSUID um die UID einer Benutzergruppe oder eines einzelnen Benutzers handelt.

Rückgabewerte:

*VARIANT_BOOL***pVal*:

0 = gehört keiner Benutzergruppe an

1 = gehört einer Benutzergruppe an

Beispiel:

```
pVal = IDrtAccessEntry.IsGroup
```

Siehe: [Allgemeine Beispiele](#)

HRESULT IsGroup ([in] VARIANT_BOOL *newVal*)

Beschreibung: Diese Methode setzt den OSUID Wert um die UID für eine Benutzergruppe oder eines einzelnen Benutzers festzulegen.

Parameter:

VARIANT_BOOL *newVal*

0 = gehört keiner Benutzergruppe an

1 = gehört einer Benutzergruppe an

Beispiel:

```
IDrtAccessEntry.IsGroup(newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ModifyIndex ([out, retval] long * *pVal*)

Beschreibung: Eigenschaft gibt an, ob der Benutzer die Indexdaten des Objektes ändern darf.

Rückgabewerte:

*LONG***pVal*: Eigenschaft ob Benutzer Objektindexdaten verändern kann

Beispiel:

```
pVal = IDrtAccessEntry.ModifyIndex
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ModifyIndex ([in] long *newVal*)

Beschreibung: Setzt Eigenschaft, ob der Benutzer die Indexdaten des Objektes ändern darf.

Parameter:

*LONG**newVal*: Setzt Eigenschaft ob Benutzer Objektindexdaten ändern kann

Beispiel:

```
IDrtAccessEntry.ModifyIndex(newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ModifyObject ([out, retval] long * *pVal*)

Beschreibung: Eigenschaft gibt an, ob der Benutzer das Dokument zum Objekt ändern darf.

Rückgabewerte:

*LONG***pVal*: Eigenschaft gibt an ob Benutzer Dokument zu Objekt ändern darf

Beispiel:

```
pVal = IDrtAccessEntry.Object
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ModifyObject ([in] long *newVal*)

Beschreibung: Setzt Eigenschaft, ob der Benutzer das Dokument zum Objekt ändern darf.

Parameter:

*LONG**newVal*: Setzt die Eigenschaft

Beispiel:

```
IDrtAccessEntry.Object(newVal)
```


Siehe: [Allgemeine Beispiele](#)

HRESULT ExportObject ([out, retval] long * *pVal*)

Beschreibung: Eigenschaft gibt an, ob der Benutzer das Objekt exportieren darf.

Rückgabewerte:

*LONG** *pVal*: Erhält die Eigenschaft

Beispiel:

```
pVal = IDrtAccessEntry.ExportObject
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ExportObject ([in] long *newVal*)

Beschreibung: Setzt Eigenschaft, ob der Benutzer das Objekt exportieren darf.

Parameter:

LONG *newVal*: Setz die Eigenschaft

Beispiel:

```
IDrtAccessEntry.ExportObject (newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT DeleteObject ([out, retval] long * *pVal*)

Beschreibung: Eigenschaft gibt an, ob der Benutzer das gesamte Objekt löschen darf.

Rückgabewerte:

*LONG** *pVal*: Erhält die Eigenschaft

Beispiel:

```
pVal = IDrtAccessEntry.DeleteObject
```

Siehe: [Allgemeine Beispiele](#)

HRESULT DeleteObject ([in] long *newVal*)

Beschreibung: Setzt die Eigenschaft, ob der Benutzer das gesamte Objekt löschen darf.

Parameter:

LONG *newVal*: Setzt die Eigenschaft

Beispiel:

```
IDrtAccessEntry.DeleteObject (newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ModifySecurity ([out, retval] long * *pVal*)

Beschreibung: Eigenschaft gibt an, ob der Benutzer das Objekt modifizieren darf.

Rückgabewerte:

*LONG**pVal: Erhält die Eigenschaft

Beispiel:

```
pVal = IDrtAccessEntry.ModifySecurity
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ModifySecurity ([in] long *newVal*)

Beschreibung: Setzt die Eigenschaft, ob der Benutzer das Objekt modifizieren darf.

Parameter:

*LONG*newVal: Setzt die Eigenschaft

Beispiel:

```
IDrtAccessEntry.ModifySecurity(newVal)
```

Siehe: [Allgemeine Beispiele](#)

IDrtAccessList Schnittstellenreferenz

Abgeleitet von IDispatch.

Ausführliche Beschreibung

Diese Schnittstelle definiert einen einzelnen SecurityDescriptor, der den Zugriff auf die DMS - Daten steuern. Er enthält eine Liste von einzelnen Zugriffssteuerungseinträgen ([IDrtAccessEntry](#)), die die einzelnen Zugriffsrechte für einen einzelnen Benutzer (/Benutzergruppe) darstellen.

Siehe auch: [IDrtAccessEntry](#)

Öffentliche Methoden

HRESULT [Deserialize](#) ([in] BSTR bsXml)
 HRESULT [Serialize](#) ([out, retval] BSTR *pbsXml)
 HRESULT [ObjectType](#) ([out, retval] long *pVal)
 HRESULT [ObjectType](#) ([in] long newVal)
 HRESULT [ObjectID](#) ([out, retval] long *pVal)
 HRESULT [ObjectID](#) ([in] long newVal)
 HRESULT [OSSD](#) ([out, retval] BSTR *pVal)
 HRESULT [OSSD](#) ([in] BSTR newVal)

Dokumentation der Elementfunktionen

HRESULT Deserialize ([in] BSTR *bsXml*)

Beschreibung: Diese Methode initialisiert ein [IDrtAccessList](#) - Objekt einschliesslich seiner Attribute und Unterobjekte aus einem XML String.

Parameter:

BSTR bsXml: xml-input string

Beispiel:

```
IDrtAccessList.Deserialize(bsXml)
```

Siehe: [Allgemeine Beispiele](#) , DMSAccess

HRESULT Serialize ([out, retval] BSTR **pbsXml*)

Beschreibung: Diese Methode serialisiert das [IDrtAccessList](#) - Objekt einschliesslich seiner Attribute und Unterobjekte vollständig in einen XML String.

Rückgabewerte:

*BSTR**pbsXml: output xml-string

Beispiel:

```
pbsXml = IDrtAccessList.Serialize
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ObjectType ([out, retval] long * *pVal*)

Beschreibung: Diese Methode erhält den Objekttypen des DMS - Datenobjektes, dem der SecurityDescriptor zugeordnet ist.

Rückgabewerte:

*LONG** *pVal*: Wert des Objekttyps

Beispiel:

```
pVal = IDrtAccessList.ObjectType
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ObjectType ([in] long *newVal*)

Beschreibung: Diese Methode setzt den Objekttypen des DMS - Datenobjektes, dem der SecurityDescriptor zugeordnet ist.

Parameter:

LONG *newVal*: Wert des Objekttyps

Beispiel:

```
IDrtAccessList.ObjectType (newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ObjectID ([out, retval] long * *pVal*)

Beschreibung: Diese Methode erhält die ID des DMS - Datenobjektes, dem der SecurityDescriptor zugeordnet ist.

Rückgabewerte:

*LONG** *pVal*: ID des DMS - Datenobjektes

Beispiel:

```
pVal = IDrtAccessList.ObjectID
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ObjectID ([in] long *newVal*)

Beschreibung: Diese Methode setzt die ID des DMS - Datenobjektes, dem der SecurityDescriptor zugeordnet ist.

Parameter:

LONG *newVal*: ID des DMS - Datenobjektes

Beispiel:

```
IDrtAccessList.ObjectID (newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT OSSD ([out, retval] BSTR * *pVal*)

Beschreibung: Diese Methode erhält die GUID des SecurityDescriptors.

Rückgabewerte:

*BSTR**pVal: GUID des SecurityDescriptors

Beispiel:

```
pVal = IDrtAccessList.OSSD
```

Siehe: [Allgemeine Beispiele](#)

HRESULT OSSD ([in] *BSTR newVal*)

Beschreibung: Diese Methode setzt die GUID des SecurityDescriptors.

Parameter:

BSTR newVal: GUID des SecurityDescriptors

Beispiel:

```
IDrtAccessList.OSSD(newVal)
```

Siehe: [Allgemeine Beispiele](#)

IDrtField Schnittstellenreferenz

Abgeleitet von IDispatch.

Ausführliche Beschreibung

Diese Schnittstelle beschreibt ein einzelnes Feldobjekt.

Öffentliche Methoden

HRESULT [Deserialize](#) ([in] BSTR bstrXml)
 HRESULT [Serialize](#) ([out, retval] BSTR *pbstrXml)
 HRESULT [System](#) ([out, retval] VARIANT_BOOL *pbSystem)
 HRESULT [Name](#) ([out, retval] BSTR *pbstrName)
 HRESULT [Name](#) ([in] BSTR bstrName)
 HRESULT [InternalName](#) ([out, retval] BSTR *pbstrInternalName)
 HRESULT [InternalName](#) ([in] BSTR bstrInternalName)
 HRESULT [GUID](#) ([out, retval] BSTR *pbstrGUID)
 HRESULT [GUID](#) ([in] BSTR bstrGUID)
 HRESULT [DbName](#) ([out, retval] BSTR *pbstrDbName)
 HRESULT [DataType](#) ([out, retval] BSTR *pbstrDataType)
 HRESULT [DataType](#) ([in] BSTR bstrDataType)
 HRESULT [Value](#) ([out, retval] BSTR *pbstrValue)
 HRESULT [Value](#) ([in] BSTR bstrValue)
 HRESULT [Type](#) ([out, retval] DrtFieldTypeEnum *pType)
 HRESULT [Type](#) ([in] DrtFieldTypeEnum Type)
 HRESULT [MultiField](#) ([out, retval] [IDrtMultiField](#) **ppMultiField)
 HRESULT [TableField](#) ([out, retval] [IDrtTableField](#) **ppTableField)

Dokumentation der Elementfunktionen

HRESULT Deserialize ([in] BSTR *bstrXml*)

Beschreibung: Diese Methode initialisiert ein [IDrtField](#) - Objekt einschliesslich seiner Attribute und Unterobjekte aus einem XML String.

Parameter:

BSTR pbstrXml XML String

Beispiel:

```
DrtField.Deserialize (pbstrXml)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Serialize ([out, retval] BSTR **pbstrXml*)

Beschreibung: Diese Methode serialisiert das [IDrtField](#) - Objekt einschliesslich seiner Attribute und Unterobjekte vollständig in einen XML String.

Rückgabewerte:

*BSTR**pbstrXml XML String

Beispiel:

```
pbstrXml = DrtField.Serialize
```

Siehe: [Allgemeine Beispiele](#)

HRESULT System ([out, retval] VARIANT_BOOL * *pbSystem*)

Beschreibung: Diese Methode überprüft ob das ein Systemfeld ist oder nicht.

Rückgabewerte:

*VARIANT_BOOL** pbSystem: pbSystem = true dann ist es ein Systemfeld, ansonsten false.

Beispiel:

```
pbSystem = DrtField.System
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Name ([out, retval] BSTR * *pbstrName*)

Beschreibung: Diese Methode erhält den Namen des Feldes.

Rückgabewerte:

*BSTR** pbstrName Name des Feldes in Form eines Strings

Beispiel:

```
pbstrName = DrtField.Name
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Name ([in] BSTR *bstrName*)

Beschreibung: Diese Methode setzt den Namen des Feldes.

Parameter:

BSTR bstrName Name des Feldes in Form eines Strings

Beispiel:

```
DrtField.Name (pbstrName)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT InternalName ([out, retval] BSTR * *pbstrInternalName*)

Beschreibung: Diese Methode erhält den internen Namen des Feldes.

Rückgabewerte:

*BSTR** pbstrInternalName interner Name des Feldes

Beispiel:

```
pbstrInternalName = DrtField.InternalName
```

Siehe: [Allgemeine Beispiele](#)

HRESULT InternalName ([in] BSTR *bstrInternalName*)

Beschreibung: Diese Methode setzt den internen Namen des Feldes.

Parameter:

BSTR bstrInternalName interner Name des Feldes

Beispiel:

```
DrtField.InternalName(pbstrInternalName)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT GUID ([out, retval] BSTR * *pbstrGUID*)

Beschreibung: Diese Methode erhält die GUID des Feldes.

Rückgabewerte:

*BSTR** pbstrGUID GUID des Feldes

Beispiel:

```
pbstrGUID = DrtField.GUID
```

Siehe: [Allgemeine Beispiele](#)

HRESULT GUID ([in] BSTR *bstrGUID*)

Beschreibung: Diese Methode setzt die GUID des Feldes.

Parameter:

BSTR bstrGUID GUID des Feldes

Beispiel:

```
DrtField.GUID(bstrGUID)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT DbName ([out, retval] BSTR * *pbstrDbName*)

Beschreibung: Diese Methode erhält den Datenbankfeldnamen.

Rückgabewerte:

*BSTR** pbstrDbName Datenbankfeldname als String

Beispiel:

```
pbstrDbName = DrtField.DbName
```

Siehe: [Allgemeine Beispiele](#)

HRESULT DataType ([out, retval] BSTR * *pbstrDataType*)

Beschreibung: Diese Methode erhält den Datentypen des Feldes.

Der String kann beispielsweise den Wert:

TEXT/DATE/INTEGER/ haben

Rückgabewerte:*BSTR**pbstrDataType String mit den Datentypen**Beispiel:**

```
pbstrDataType = DrtField.DataType
```

Siehe: [Allgemeine Beispiele](#)

HRESULT DataType ([in] BSTR *bstrDataType*)

Beschreibung: Diese Methode setzt den Datentypen des Feldes.

Parameter:*BSTR* bstrDataTypeString mit den Datentypen des Feldes.**Beispiel:**

```
DrtField.DataType(bstrDataType)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Value ([out, retval] BSTR * *pbstrValue*)

Beschreibung: Diese Methode erhält den Wert des Feldes.

Rückgabewerte:*BSTR**pbstrValue Wert des Feldes**Beispiel:**

```
pbstrValue = DrtField.Value
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Value ([in] BSTR *bstrValue*)

Beschreibung: Diese Methode setzt den Wert des Feldes.

Parameter:*BSTR* bstrValue Wert des Feldes**Beispiel:**

```
DrtField.Value(bstrValue)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Type ([out, retval] DrtFieldTypeEnum * *pType*)

Beschreibung: Diese Methode gibt den Feldtypen des Feldes zurück (einfaches, Mehrfach- oder Tabellenfeld).

Der Parameter kann den Wert:

DrtFieldType_SimpleField = 0

DrtFieldType_MultiField = 2

DrtFieldType_TableField = 2 haben

Rückgabewerte:

*DrtFieldTypeEnum** pType Wert für den Feldtyp

Beispiel:

```
pType = DrtField.Type
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Type ([in] DrtFieldTypeEnum Type)

Beschreibung: Diese Methode setzt den Feldtypen des Feldes (einfaches, Mehrfach- oder Tabellenfeld).

Der Parameter kann den Wert:

DrtFieldType_SimpleField = 0
DrtFieldType_MultiField = 2
DrtFieldType_TableField = 2 haben

Parameter:

DrtFieldTypeEnum Type Wert für den Feldtyp

Beispiel:

```
DrtField.Type (DrtFieldTypeEnum Type)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT MultiField ([out, retval] [IDrtMultiField](#) ** ppMultiField)

Beschreibung: Diese Methode erhält das Mehrfachfeld - Objekt des Feldes, sofern das Feld ein Mehrfachfeld darstellt. Ist dies nicht der Fall wird der Rückgabewert auf 0 gesetzt.

Rückgabewerte:

*IDrtMultifield*** ppMultifield

Beispiel:

```
ppMultifield = DrtField.Multifield
```

Siehe: [Allgemeine Beispiele](#)

HRESULT TableField ([out, retval] [IDrtTableField](#) ** ppTableField)

Beschreibung: Diese Methode erhält das Tabellenfeld - Objekt des Feldes, sofern das Feld ein Tabellenfeld darstellt. Ist dies nicht der Fall wird der Rückgabewert auf 0 gesetzt.

Rückgabewerte:

*IDrtTableField*** ppTableField

Beispiel:

```
ppTableField = DrtField.TableField
```

Siehe: [Allgemeine Beispiele](#)

IDrtFieldConditionNode Schnittstellenreferenz

Abgeleitet von IDispatch.

Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Bedingung auf einen Indexdatenfeld, die zur Recherche nach DMS - Objekten verwendet werden kann.

Öffentliche Methoden

HRESULT [Deserialize](#) ([in] BSTR bstrXml)
 HRESULT [Serialize](#) ([out, retval] BSTR *pbstrXml)
 HRESULT [NodeType](#) ([out, retval] DrtConditionNodeTypeEnum *pNodeType)
 HRESULT [NodeType](#) ([in] DrtConditionNodeTypeEnum NodeType)
 HRESULT [Field](#) ([out, retval] [IDrtField](#) **ppField)
 HRESULT [Field](#) ([in] [IDrtField](#) *pField)
 HRESULT [Operator](#) ([out, retval] DrtOperatorEnum *pOperator)
 HRESULT [Operator](#) ([in] DrtOperatorEnum Operator)
 HRESULT [GetValueCount](#) ([out, retval] long *plCount)
 HRESULT [GetValue](#) ([in] long lIndex, [out, retval] BSTR *pbstrValue)
 HRESULT [AddValue](#) ([in] BSTR bstrValue, [out, retval] long *plIndex)
 HRESULT [ClearValues](#) ()

Dokumentation der Elementfunktionen

HRESULT Deserialize ([in] BSTR *bstrXml*)

Beschreibung: Diese Methode setzt die Objekteigenschaften anhand einer XML Beschreibung.

Parameter:

BSTR bstrXml: String mit der XML Beschreibung

Beispiel:

```
DrtFieldConditionNode.Deserialize(BSTR bstrXml)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Serialize ([out, retval] BSTR **pbstrXml*)

Beschreibung: Diese Methode erhält als Rückgabewert die Objekteigenschaften in Form eine XML Beschreibung.

Rückgabewerte:

*BSTR** pbstrXml: String mit der XML Beschreibung

Beispiel:

```
BSTR* pbstrXml = DrtFieldConditionNode.Serialize
```

Siehe: [Allgemeine Beispiele](#)

HRESULT NodeType ([out, retval] DrtConditionNodeTypeEnum * pNodeType)

Beschreibung: Der Knotentyp wird zurück gegeben.

Mögliche Werte:

DrtConditionNodeType_Leaf = 0: Defaultwert

DrtConditionNodeType_Or = 1

DrtConditionNodeType_And = 2

Rückgabewerte:

DrtConditionNodeTypeEnum pNodeType*

Der Knotentyp gibt an, ob der Knoten ein eine einzelne Feldbedingung (Leaf) ist oder als Liste von mehreren logisch verknüpften Feldbedingungen dient. Ist das Objekt eine Bedingungsliste, so sind die einzelnen Bedingungen dieser Liste immer entweder ODER oder UND verknüpft. Je nach Typ des Listenknotens.

Beispiel:

```
DrtConditionNodeTypeEnum*pNodeType = DrtFieldConditionNode.NodeType
```

Siehe: [Allgemeine Beispiele](#)

HRESULT NodeType ([in] DrtConditionNodeTypeEnum NodeType)

Beschreibung: Der Knotentyp wird gesetzt.

Mögliche Werte:

DrtConditionNodeType_Leaf = 0: Defaultwert

DrtConditionNodeType_Or = 1

DrtConditionNodeType_And = 2

Rückgabewerte:

DrtConditionNodeTypeEnum pNodeType*

Der Knotentyp gibt an, ob der Knoten ein eine einzelne Feldbedingung (Leaf) ist oder als Liste von mehreren logisch verknüpften Feldbedingungen dient. Ist das Objekt eine Bedingungsliste, so sind die einzelnen Bedingungen dieser Liste immer entweder ODER oder UND verknüpft. Je nach Typ des Listenknotens.

Beispiel:

```
DrtFieldConditionNode.NodeType = DrtConditionNodeType_Or
DrtFieldConditionNode.Add 0,FieldCond1           // Zuweisung weiterer
Knoten vom Typ DrtConditionNodeType_Leaf
DrtFieldConditionNode.Add 1,FieldCond2           // Zuweisung weiterer
Knoten vom Typ DrtConditionNodeType_Leaf
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Field ([out, retval] IDrtField ** ppField)

Beschreibung: Das DrtField-Objekt definiert das Feld, dessen Wert geprüft werden soll.

Rückgabewerte:

*IDrtField** ppField*

Beispiel:

```
IDrtField**ppField = DrtFieldConditionNode.Field
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Field ([in] [IDrtField](#) * *pField*)

Beschreibung: Das DrtField-Objekt definiert das Feld, dessen Wert geprüft werden soll.

Parameter:

*IDrtField** pField: gibt das Feldobjekt zurück

Beispiel:

```
DrtFieldConditionNode.Field(IDrtField*pField)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Operator ([out, retval] DrtOperatorEnum * *pOperator*)

Beschreibung: Der Vergleichsoperator aus der DrtOperatorEnum-Enumeration für den Vergleich eines Feldwertes mit einem konstanten Wert.

Mögliche Werte:

DrtOperatorEnum_Equ = 0 - gleich (A = B)
 DrtOperatorEnum_Neq = 1 - ungleich (A != B)
 DrtOperatorEnum_Les = 2 - kleiner (A < B)
 DrtOperatorEnum_Leq = 3 - kleiner gleich (A <= B)
 DrtOperatorEnum_Gre = 4 - größer (A > B)
 DrtOperatorEnum_Geq = 5 - größer gleich (A >= B)
 DrtOperatorEnum_Bet = 6 - liegt zwischen (A BETWEEN B,C)
 DrtOperatorEnum_Nbe = 7 - liegt nicht zwischen (A BETWEEN B,C)
 DrtOperatorEnum_In = 8 - enthalten in (A IN B)
 DrtOperatorEnum_Nin = 9 - nicht enthalten in (A NOT IN B)

Rückgabewerte:

*DrtOperatorEnum** pOperator

Beispiel:

```
DrtOperatorEnum*pOperator = DrtFieldConditionNode.Operator
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Operator ([in] DrtOperatorEnum *Operator*)

Beschreibung: Der Vergleichsoperator aus der DrtOperatorEnum-Enumeration für den Vergleich eines Feldwertes mit einem konstanten Wert.

Mögliche Werte:

DrtOperatorEnum_Equ = 0 - gleich (A = B)
 DrtOperatorEnum_Neq = 1 - ungleich (A != B)
 DrtOperatorEnum_Les = 2 - kleiner (A < B)
 DrtOperatorEnum_Leq = 3 - kleiner gleich (A <= B)
 DrtOperatorEnum_Gre = 4 - größer (A > B)
 DrtOperatorEnum_Geq = 5 - größer gleich (A >= B)
 DrtOperatorEnum_Bet = 6 - liegt zwischen (A BETWEEN B,C)
 DrtOperatorEnum_Nbe = 7 - liegt nicht zwischen (A BETWEEN B,C)
 DrtOperatorEnum_In = 8 - enthalten in (A IN B)
 DrtOperatorEnum_Nin = 9 - nicht enthalten in (A NOT IN B)

Parameter:

DrtOperatorEnum Operator

Beispiel:

```
DrtFieldConditionNode.Operator(DrtOperatorEnum Operator)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT GetValueCount ([out, retval] long * *plCount*)

Beschreibung: Liefert die Anzahl der konstanten Werte für den Vergleich. Die zulässige Anzahl hängt vom Operator ab: Die meisten Operatoren brauchen 1 Wert, (NOT) BETWEEN braucht 2 Werte und (NOT) IN 1..n.

Rückgabewerte:

*LONG** *plCount*

Beispiel:

```
LONG* plCount = DrtFieldConditionNode.GetValueCount
```

Siehe: [Allgemeine Beispiele](#)

HRESULT GetValue ([in] long *lIndex*, [out, retval] BSTR * *pbstrValue*)

Beschreibung: Liefert den konstanten Wert mit dem angegebenen Index (0-basiert).

Parameter:

LONG *lIndex*

Rückgabewerte:

*BSTR** *pbstrValue*

Beispiel:

```
BSTR* pbstrValue = DrtFieldConditionNode.GetValue (LONG lIndex)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT AddValue ([in] BSTR *bstrValue*, [out, retval] long * *plIndex*)

Beschreibung: Fügt einen Vergleichswert hinzu.

Parameter:

BSTR *bstrValue*

Rückgabewerte:

*LONG** *plIndex*

Beispiel:

```
LONG* plIndex = DrtFieldConditionNode.AddValue (BSTR bstrValue)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ClearValues ()

Beschreibung: Entfernt alle bereits eingefügten Vergleichswerte.

Beispiel:

```
DrtFieldConditionNode.ClearValues()
```

Siehe: [Allgemeine Beispiele](#)

IDrtFields Schnittstellenreferenz

Abgeleitet von IDispatch.

Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Zusammenstellung von Objekten vom Typ [IDrtField](#).

Siehe auch: [IDrtField](#)

Öffentliche Methoden

HRESULT [Deserialize](#) ([in] BSTR bstrXml)
HRESULT [Serialize](#) ([out, retval] BSTR *pbstrXml)
HRESULT [ItemByName](#) ([in] BSTR bstrName,[out, retval] [IDrtField](#) **ppField)
HRESULT [ItemByInternalName](#) ([in] BSTR bstrInternalName,[out, retval] [IDrtField](#) **ppField)
HRESULT [ItemByGUID](#) ([in] BSTR bstrGUID,[out, retval] [IDrtField](#) **ppField)

Dokumentation der Elementfunktionen

HRESULT Deserialize ([in] BSTR *bstrXml*)

Beschreibung Diese Methode beschreibt eine Deserialisierung einer Felder - Liste aus einem XML String.

Parameter:

BSTR strXml: String der das XML enthält, aus dem die Feld-Collection initialisiert werden soll.

Beispiel:

```
DrtObject.Deserialize(BSTR strXml)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Serialize ([out, retval] BSTR **pbstrXml*)

Beschreibung Diese Methode beschreibt eine Serialisierung einer Felder - Liste in XML.

Rückgabewerte:

*BSTR**pbstrXml: String, der die XML Entsprechung der Feld-Collection enthält

Beispiel:

```
BSTR*strXml = DrtObject.Serialize
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ItemByName ([in] BSTR *bstrName*, [out, retval] [IDrtField](#) ***ppField*)

Beschreibung Diese Methode gibt ein Element der Felder - Liste nach dem Namen zurück.

Parameter:

BSTR bstrName

Rückgabewerte:*IDrtField*** pField: Feldobjekt**Beispiel:**

```
IDrtField**ppField = DrtObject.ItemByName(BSTR bstrName)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ItemByInternalName ([in] BSTR *bstrInternalName*, [out, retval] [IDrtField](#) ** *ppField*)

Beschreibung Diese Methode gibt ein Element der Felder - Liste nach dem internen Namen zurück.

Parameter:*BSTR* bstrInternalName**Rückgabewerte:***IDrtField*** pField**Beispiel:**

```
IDrtField**pField = DrtObject.ItemByName(BSTR bstrInternalName)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ItemByGUID ([in] BSTR *bstrGUID*, [out, retval] [IDrtField](#) ** *ppField*)

Beschreibung Diese Methode gibt ein Element der Felder - Liste nach der GUID zurück.

Parameter:*BSTR* bstrGUID**Rückgabewerte:***IDrtField*** pField**Beispiel:**

```
IDrtField**pField = DrtObject.ItemByGUID(BSTR bstrGUID)
```

IDrtGroup Schnittstellenreferenz

Abgeleitet von IDispatch.

Ausführliche Beschreibung

Diese Schnittstelle bietet den Zugriff auf die Daten einer Benutzergruppe im System.

Öffentliche Methoden

HRESULT [Deserialize](#) ([in] BSTR bsXml)
HRESULT [Serialize](#) ([out, retval] BSTR *pbsXml)
HRESULT [Description](#) ([out, retval] BSTR *pVal)
HRESULT [Description](#) ([in] BSTR newVal)
HRESULT [ID](#) ([out, retval] long *pVal)
HRESULT [ID](#) ([in] long newVal)
HRESULT [Name](#) ([out, retval] BSTR *pVal)
HRESULT [Name](#) ([in] BSTR newVal)
HRESULT [GUID](#) ([out, retval] BSTR *pVal)
HRESULT [GUID](#) ([in] BSTR newVal)
HRESULT [Profil](#) ([out, retval] long *pVal)
HRESULT [Profil](#) ([in] long newVal)

Dokumentation der Elementfunktionen

HRESULT Deserialize ([in] BSTR *bsXml*)

Beschreibung: Diese Methode initialisiert eine Benutzergruppe aus einem XML formatierten String.

Parameter:

BSTR bsXml: xml-input string

Beispiel:

```
IDrtGroup.Deserialize(bsXml)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Serialize ([out, retval] BSTR **pbsXml*)

Beschreibung: Diese Methode serialisiert die Eigenschaften einer Benutzergruppe in einen XML String.

Rückgabewerte:

*BSTR**pbsXml: output xml-string

Beispiel:

```
pbsXml = IDrtGroup.Serialize
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Description ([out, retval] BSTR **pVal*)

Bechreibung: Diese Methode liefert den Beschreibungstext zu der Benutzergruppe.

Rückgabewerte:

*BSTR**pVal: Beschreibungstext als String

Beispiel:

```
pVal = IDrtGroup.Description
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Description ([in] *BSTR newVal*)

Beschreibung: Diese Methode setzt einen neuen Beschreibungstext für die aktuelle Benutzergruppe.

Parameter:

BSTR newVal: neuer Beschreibungstext als String

Beispiel:

```
IDrtGroup.Description(newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ID ([out, retval] long * *pVal*)

Beschreibung: Diese Methode liefert die Benutzergruppen id.

Rückgabewerte:

*LONG**pVal

Beispiel:

```
pVal = IDrtGroup.ID
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ID ([in] long *newVal*)

Beschreibung: Diese Methode setzt die Benutzergruppen id.

Parameter:

LONG newVal

Beispiel:

```
IDrtGroup.ID(newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Name ([out, retval] *BSTR * pVal*)

Beschreibung: Diese Methode liefert den Namen der Benutzergruppe.

Rückgabewerte:

*BSTR**pVal

Beispiel:

```
pVal = IDrtGroup.Name
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Name ([in] BSTR *newVal*)

Beschreibung: Diese Methode setzt den Namen der Benutzergruppe.

Parameter:

BSTR newVal

Beispiel:

```
IDrtGroup.Name(newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT GUID ([out, retval] BSTR * *pVal*)

Beschreibung: Diese Methode liefert die GUID von der identifizierten Benutzergruppe.

Rückgabewerte:

*BSTR** pVal

Beispiel:

```
pVal = IDrtGroup.GUID
```

Siehe: [Allgemeine Beispiele](#)

HRESULT GUID ([in] BSTR *newVal*)

Beschreibung: Diese Methode setzt die GUID von der identifizierten Benutzergruppe.

Parameter:

BSTR newVal

Beispiel:

```
IDrtGroup.GUID(newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Profil ([out, retval] long * *pVal*)

Beschreibung: Diese Methode liefert die Benutzergruppen - Profil - ID.

Rückgabewerte:

*LONG** pVal

Beispiel:

```
pVal = IDrtGroup.Profil
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Profil ([in] long *newVal*)

Beschreibung: Diese Methode setzt die Benutzergruppen - Profil - ID.

Parameter:*LONG*newVal**Beispiel:**

```
IDrtGroup.Profil(newVal)
```

Siehe: [Allgemeine Beispiele](#)

IDrtGroups Schnittstellenreferenz

Abgeleitet von IDispatch.

Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Zusammenstellung von Objekten vom Typ [IDrtGroup](#).

Siehe auch: [IDrtGroup](#)

Öffentliche Methoden

HRESULT [Deserialize](#) ([in] BSTR bsXml)

HRESULT [Serialize](#) ([out, retval] BSTR *pbsXml)

Dokumentation der Elementfunktionen

HRESULT Deserialize ([in] BSTR *bsXml*)

Beschreibung: Diese Methode initialisiert ein [IDrtGroup](#) - Objekt einschliesslich seiner Attribute und Unterobjekte aus einem XML String.

Parameter:

BSTR bsXml: xml-input string

Beispiel:

```
IDrtGroups.Deserialize(bsXml)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Serialize ([out, retval] BSTR **pbsXml*)

Beschreibung: Diese Methode serialisiert das [IDrtGroup](#) - Objekt einschliesslich seiner Attribute und Unterobjekte vollständig in einen XML String.

Rückgabewerte:

*BSTR**pbsXml: output xml-string

Beispiel:

```
pbsXml = IDrtGroups.Serialize
```

Siehe: [Allgemeine Beispiele](#)

IDrtMultiField Schnittstellenreferenz

Abgeleitet von IDispatch.

Ausführliche Beschreibung

Diese Schnittstelle beschreibt ein spezielles Feldobjekt: Mehrfachfeld

Öffentliche Methoden

HRESULT [GetPageCount](#) ([out, retval] long *plCount)
HRESULT [GetPageNumber](#) ([in] long lPageIndex,[out, retval] long *plPageNumber)
HRESULT [GetPageEntryCount](#) ([in] long lPageIndex,[out, retval] long *plCount)
HRESULT [AddPage](#) ([in] long lPageNumber,[out, retval] long *plPageIndex)
HRESULT [GetEntry](#) ([in] long lPageIndex,[in] long lEntryIndex,[out, retval] BSTR *pbstrEntry)
HRESULT [AddEntry](#) ([in] long lPageIndex,[in] BSTR bstrValue,[out, retval] long *plEntryIndex)
HRESULT [Clear](#) (void)
HRESULT [ClearPage](#) (long lPageNumber)

Dokumentation der Elementfunktionen

HRESULT GetPageCount ([out, retval] long *plCount)

Beschreibung: Diese Methode erhält die Anzahl der Seiten eines Mehrfachfeldes.

Rückgabewerte:

*LONG**plCount Anzahl der Seiten vom Mehrfachfeld

Beispiel:

```
plCount = DrtMultiField.GetPageCount
```

Siehe: [Allgemeine Beispiele](#)

HRESULT GetPageNumber ([in] long lPageIndex, [out, retval] long *plPageNumber)

Beschreibung: Diese Methode gibt die Seitennummer der durch den Index angegebenen Seite eines Mehrfachfeldes zurück.

Parameter:

*LONG*lPageIndex: Index der aktuellen Seite.

Rückgabewerte:

*LONG**plPageNumber: Seitennummer

Beispiel:

```
plPageNumber = DrtMultiField.GetPageNumber(lPageIndex)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT GetPageEntryCount ([in] long lPageIndex, [out, retval] long *plCount)

Beschreibung: Diese Methode erhält die Anzahl der Felder auf einer Seite eines Mehrfachfeldes.

Parameter:

LONG lPageIndex: Index der aktuellen Seite.

Rückgabewerte:

*LONG** plCount: Anzahl der Felder

Beispiel:

```
plCount = DrtMultiField.GetPageEntryCount(lPageIndex)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT AddPage ([in] long lPageNumber, [out, retval] long * plPageIndex)

Beschreibung: Diese Methode legt eine neue Seiten in einem Mehrfachfeld an und gibt den den Index der Seite zurück.

Parameter:

LONG lPageNumber: Seitennummer der neuen Seite.

Rückgabewerte:

*LONG** plPageIndex: Index der eingefügten Seite

Beispiel:

```
plPageIndex = DrtMultiField.AddPage(lPageNumber)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT GetEntry ([in] long lPageIndex, [in] long lEntryIndex, [out, retval] BSTR * pbstrEntry)

Beschreibung: Diese Methode gibt den Wert eines Feldes auf einer Seite eines Mehrfachfeldes zurück.

Parameter:

LONG lPageIndex: Index der aktuellen Seite.

LONG lEntryIndex: Index des aktuellen Feldwertes.

Rückgabewerte:

*BSTR** pbstrEntry: String, der den Wert des Feldes enthält

Beispiel:

```
pbstrEntry = DrtMultiField.GetEntry(lPageIndex, lEntryIndex)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT AddEntry ([in] long lPageIndex, [in] BSTR bstrValue, [out, retval] long * plEntryIndex)

Beschreibung: Diese Methode fügt einen Feldwert auf einer Seite eines Mehrfachfeldes ein.

Parameter:

LONG lPageIndex: Index der aktuellen Seite.

BSTR bstrValue: Einzufügender Feldwert.

Rückgabewerte:

*LONG** plEntryIndex: Index des eingefügten Feldwertes.

Beispiel:

```
plEntryIndex = DrtMultiField.AddEntry(lPageIndex, bstrValue)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Clear (void)

Beschreibung: Diese Methode entfernt alle Seiten und Einträge eines Mehrfachfeldes.

Beispiel:

```
DrtMultiField.Clear
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ClearPage (long lPageNumber)

Beschreibung: Diese Methode entfernt alle Einträge auf der angegebenen Seite eines Mehrfachfeldes.

Parameter:

*LONG*lPageNumber: Seitennummer wo alle Einträge entfernt werden sollen

Beispiel:

```
DrtMultiField.ClearPage (lPageId)
```

Siehe: [Allgemeine Beispiele](#)

IDrtObject Schnittstellenreferenz

Abgeleitet von IDispatch.

Ausführliche Beschreibung

Diese Schnittstelle beschreibt ein einzelnes Indexdatenobjekt. Das Objekt enthält die Felder und Werte, mit denen das Objekt aufgrund seines Objekttyps initialisiert wurden ist.

Öffentliche Methoden

HRESULT [Deserialize](#) ([in] BSTR bstrXml)
 HRESULT [Serialize](#) ([out, retval] BSTR *pbstrXml)
 HRESULT [Reload](#) ([in] [IDrtSession](#) *pSession)
 HRESULT [ReloadAll](#) ([in] [IDrtSession](#) *pSession)
 HRESULT [Update](#) ([in] [IDrtSession](#) *pSession)
 HRESULT [Fields](#) ([out, retval] [IDrtFields](#) **ppFields)
 HRESULT [ID](#) ([out, retval] long *pID)
 HRESULT [ID](#) ([in] long IID)
 HRESULT [Name](#) ([out, retval] BSTR *pbstrName)
 HRESULT [Name](#) ([in] BSTR bstrName)
 HRESULT [InternalName](#) ([out, retval] BSTR *pbstrInternalName)
 HRESULT [InternalName](#) ([in] BSTR bstrInternalName)
 HRESULT [GUID](#) ([out, retval] BSTR *pbstrGUID)
 HRESULT [GUID](#) ([in] BSTR bstrGUID)
 HRESULT [DbName](#) ([out, retval] BSTR *pbstrDbName)
 HRESULT [Type](#) ([out, retval] DrtObjectTypeEnum *pType)
 HRESULT [Type](#) ([in] DrtObjectTypeEnum Type)
 HRESULT [FolderID](#) ([out, retval] long *pIID)
 HRESULT [FolderID](#) ([in] long IID)
 HRESULT [RegisterID](#) ([out, retval] long *pIID)
 HRESULT [RegisterID](#) ([in] long IID)
 HRESULT [CabinetID](#) ([out, retval] long *pIID)
 HRESULT [CabinetID](#) ([in] long IID)
 HRESULT [ObjectTypeID](#) ([out, retval] long *pIType)
 HRESULT [ObjectTypeID](#) ([in] long IType)
 HRESULT [MainType](#) ([out, retval] long *pIType)
 HRESULT [CoType](#) ([out, retval] long *pIType)
 HRESULT [LoadFiles](#) ([in] [IDrtSession](#) *pSession, [in] DrtLoadObjFileFlagsEnum enFlags, [out, retval] BSTR *pbstrFilenames)
 HRESULT [SaveFiles](#) ([in] [IDrtSession](#) *pSession, [in] DrtSaveObjFileFlagsEnum enFlags, [in] BSTR bstrFilenames, [out, retval] long *pISize)
 HRESULT [GetFilename](#) ([in] long lIndex, [out, retval] BSTR *pbstrFilename)
 HRESULT [AddFilename](#) ([in] BSTR bstrFilename, [out, retval] long *pISize)
 HRESULT [GetFilenames](#) ([out, retval] BSTR *pbstrFiles)
 HRESULT [SetFilenames](#) ([in] BSTR bstrFiles, [out, retval] long *pISize)
 HRESULT [InsertFilename](#) ([in] long lIndex, [in] BSTR bstrFilename, [out, retval] long *pIIndex)
 HRESULT [Childs](#) ([out, retval] [IDrtObjects](#) **ppObjects)
 HRESULT [UndoCheckout](#) ([in] [IDrtSession](#) *)
 HRESULT [FileCount](#) ([out, retval] long *pICount)
 HRESULT [FileSize](#) ([out, retval] long *pISize)
 HRESULT [FileExt](#) ([out, retval] BSTR *pbstrExt)
 HRESULT [FileMimeType](#) ([out, retval] BSTR *pbstrExt)
 HRESULT [Restore](#) ([in] [IDrtSession](#) *pSession)
 HRESULT [IsDeleted](#) ([out, retval] long *pIIsDeleted)

Dokumentation der Elementfunktionen

HRESULT Deserialize ([in] BSTR *bstrXml*)

Beschreibung: Diese Methode initialisiert ein [IDrtObject](#) - Objekt einschliesslich seiner Attribute und Unterobjekte aus einem XML String.

Parameter:

BSTR bstrXml String der das XML enthält, aus dem das Objekt initialisiert werden soll.

Beispiel:

```
DrtObject.Deserialize(bstrXml)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Serialize ([out, retval] BSTR **pbstrXml*)

Beschreibung: Diese Methode serialisiert das [IDrtObject](#) - Objekt einschliesslich seiner Attribute und Unterobjekte vollständig in einen XML String.

Rückgabewerte:

*BSTR** pbstrXml String, der die XML Entsprechung des Objektes enthält

Beispiel:

```
bstrXml = DrtObject.Serialize
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Reload ([in] [IDrtSession](#) * *pSession*)

Beschreibung: Diese Methode bewirkt ein neu Einlesen vom Objekt aus der Datenbank.

Parameter:

*IDrtSession** pSession Zeiger auf das IDrtSession-Objekt das die Verbindung zum Server hält

Beispiel:

```
Dim pSession  
Set pSession = CreateObject("axcdl.DrtSession")  
DrtObject.Reload(pSession)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ReloadAll ([in] [IDrtSession](#) * *pSession*)

Beschreibung: Diese Methode bewirkt ein neu Einlesen aller Objekte aus der Datenbank.

Parameter:

*IDrtSession** pSession Zeiger auf das IDrtSession-Objekt das die Verbindung zum Server hält

Beispiel:

```
Dim pSession  
Set pSession = CreateObject("axcdl.DrtSession")  
DrtObject.ReloadAll(pSession)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Update ([in] [IDrtSession](#) * *pSession*)

Beschreibung: Diese Methode vollzieht ein Update der Datenbank anhand des ihm zugeteilten Objekts.

Parameter:

*IDrtSession** *pSession* Zeiger auf das IDrtSession-Objekt das die Verbindung zum Server hält

Beispiel:

```
Dim pSession
Set pSession = CreateObject("axcdl.DrtSession")
DrtObject.Update(pSession)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Fields ([out, retval] [IDrtFields](#) ** *ppFields*)

Beschreibung: Diese Methode gibt einen Zeiger auf das interne Objekt [IDrtFields](#) zurück, das eine Collection von Feldern des Objektes enthält.

Rückgabewerte:

*IDrtFields*** *ppFields*

Beispiel:

```
IDrtFields iDrtFields = DrtObject.Fields
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ID ([out, retval] long * *pID*)

Beschreibung: Diese Methode erhält die Datenbank - ID des Objektes.

Rückgabewerte:

*LONG** *pID* ID des Objektes

Beispiel:

```
pID = DrtObject.ID
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ID ([in] long *IID*)

Beschreibung: Diese Methode erstellt die ID des Objektes in der Datenbank.

Parameter:

*LONG**IID* ID des Objektes welche in die DB gespeichert wird

Beispiel:

```
DrtObject.ID(IID)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Name ([out, retval] BSTR * *pbstrName*)

Beschreibung: Diese Methode gibt den Namen des Objektes zurück.

Rückgabewerte:

*BSTR**pbstrName Name des Objektes

Beispiel:

```
pbstrName = DrtObject.Name
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Name ([in] BSTR *bstrName*)

Beschreibung: Diese Methode setzt den Namen des Objektes.

Parameter:

BSTR bstrName Enthält den Namen des Objektes

Beispiel:

```
DrtObject.Name(pbstrName)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT InternalName ([out, retval] BSTR * *pbstrInternalName*)

Beschreibung: Diese Methode erhält den internen Namen des Objektes.

Rückgabewerte:

*BSTR**pbstrInternalName Erhält den internen Namen als String

Beispiel:

```
pbstrInternalName = DrtObject.InternalName
```

Siehe: [Allgemeine Beispiele](#)

HRESULT InternalName ([in] BSTR *bstrInternalName*)

Beschreibung: Diese Methode setzt den internen Namen des Objektes.

Parameter:

BSTR bstrInternalName Setzt den internen Namen

Beispiel:

```
DrtObject.InternalName(bstrInternalName)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT GUID ([out, retval] BSTR * *pbstrGUID*)

Beschreibung: Diese Methode erhält die GUID des Objektes.

Rückgabewerte:

*BSTR**pbstrGUID Erhält die GUID als String

Beispiel:

```
pbstrGUID = DrtObject.GUID
```

Siehe: [Allgemeine Beispiele](#)

HRESULT GUID ([in] BSTR *bstrGUID*)

Beschreibung Diese Methode setzt die GUID des Objektes.

Parameter:

BSTR bstrGUID Setzt die GUID als String

Beispiel:

```
DrtObject.GUID(bstrGUID)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT DbName ([out, retval] BSTR **pbstrDbName*)

Beschreibung Diese Methode gibt den Namen der Datenbanktabelle des Objektes an.

Rückgabewerte:

*BSTR**pbstrDbName Erhält die Datenbanktabelle für das Objekt

Beispiel:

```
pbstrDbName = DrtObject.DbName
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Type ([out, retval] DrtObjectTypeEnum **pType*)

Beschreibung Diese Methode erhält den Dokumenttyp des Objektes als Enum (Folder, Register, Document) an.

Mögliche Werte:

DrtObjectType_Folder = 0

DrtObjectType_Register = 1

DrtObjectType_Document = 2

Rückgabewerte:

*DrtObjectTypeEnum** pType Wert des Dokumenttyps

Beispiel:

```
pType = DrtObject.Type
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Type ([in] DrtObjectTypeEnum *Type*)

Beschreibung Diese Methode setzt den Dokumenttyp des Objektes als Enum (Folder, Register, Document).

Mögliche Werte:

DrtObjectType_Folder = 0

DrtObjectType_Register = 1

DrtObjectType_Document = 2

Parameter:

DrtObjectTypeEnum Type Wert des Dokumenttyps

Beispiel:

```
DrtObject.Type(pType)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT FolderID ([out, retval] long * *pIID*)

Beschreibung Diese Methode erhält die ID des Ordners, in dem sich das Objekt befindet.

Rückgabewerte:

*LONG** *pIID* ID des Ordners wo sich das Objekt befindet

Beispiel:

```
pIID = DrtObject.FolderID
```

Siehe: [Allgemeine Beispiele](#)

HRESULT FolderID ([in] long *IID*)

Beschreibung Diese Methode gibt die ID des Ordners an, in dem sich das Objekt befindet.

Parameter:

LONG *IID* ID des Ordners wo sich das Objekt befindet

Beispiel:

```
DrtObject.FolderID(IID)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT RegisterID ([out, retval] long * *pIID*)

Beschreibung Diese Methode erhält die ID des Registers, in dem sich das Objekt befindet an.

Rückgabewerte:

*LONG** *pIID* Registerid wo sich das Objekt befindet

Beispiel:

```
pIID = DrtObject.RegisterID
```

Siehe: [Allgemeine Beispiele](#)

HRESULT RegisterID ([in] long *IID*)

Beschreibung Diese Methode setzt die ID des Registers, in dem sich das Objekt befindet.

Parameter:

LONG *IID* Setzt die ID des Registers wo sich das Objekt befindet

Beispiel:

```
DrtObject.RegisterID(IID)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT CabinetID ([out, retval] long * *pIID*)

Beschreibung Diese Methode erhält die ID des Schrankes, in dem sich das Objekt befindet.

Rückgabewerte:

*LONG**pIID ID des Schrankes wo sich das Objekt befindet

Beispiel:

```
pIID = DrtObject.CabinetID
```

Siehe: [Allgemeine Beispiele](#)

HRESULT CabinetID ([in] long *IID*)

Beschreibung Diese Methode setzt die ID des Schrankes, in dem sich das Objekt befindet.

Parameter:

*LONG*IID Setzt die ID des Schrankes wo sich das Objekt befindet

Beispiel:

```
DrtObject.CabinetID(LONG IID)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ObjectTypeID ([out, retval] long * *pIType*)

Beschreibung Diese Methode erhält den Objekttyp des Objektes.

Rückgabewerte:

*LONG**pIType Objekttyp des Objektes

Beispiel:

```
pIType = DrtObject.ObjectTypeID
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ObjectTypeID ([in] long *IType*)

Beschreibung Diese Methode setzt den Objekttyp des Objektes.

Parameter:

*LONG*IType Objekttyp des Objektes

Beispiel:

```
DrtObject.ObjectTypeID(IType)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT MainType ([out, retval] long * *pIType*)

Beschreibung Diese Methode erhält den Haupttyp des Objektes.

Rückgabewerte:*LONG**plType Haupttyp des Objektes**Beispiel:**

```
plType = DrtObject.MainType
```

Siehe: [Allgemeine Beispiele](#)

HRESULT CoType ([out, retval] long * *plType*)

Beschreibung Diese Methode setzt den Cotyp des Objektes.

Rückgabewerte:*LONG**plType CoTyp des Objektes**Beispiel:**

```
plType = DrtObject.CoType
```

Siehe: [Allgemeine Beispiele](#)

HRESULT LoadFiles ([in] [IDrtSession](#) * *pSession*, [in] DrtLoadObjFileFlagsEnum *enFlags*, [out, retval] BSTR * *pbstrFilenames*)

Beschreibung Lädt die Dateien des Objektes vom Server.

Parameter:*IDrtSession** pSession Zeiger auf das IDrtSession-Objekt*DrtLoadObjFileFlagsEnum* enFlags Flags für das Laden der Dateien ()**Rückgabewerte:***BSTR**bsFilenames Liste mit Dateinamen (Komma separierte Liste der Dateinamen mit Pfaden)

```
Dim pSession
Set pSession = CreateObject("axcdl.DrtSession")
bsFilenames = DrtObject.LoadFiles(pSession, 0)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT SaveFiles ([in] [IDrtSession](#) * *pSession*, [in] DrtSaveObjFileFlagsEnum *enFlags*, [in] BSTR *bstrFilenames*, [out, retval] long * *plSize*)

Beschreibung Speichert die Dateien des Objektes auf dem Server.

Parameter:*IDrtSession** pSession Zeiger auf das IDrtSession-Objekt*DrtSaveObjFileFlagsEnum* enFlags Flags für das Speichern der Dateien ()**Rückgabewerte:***BSTR**bsFilenames Liste mit Dateinamen (Komma separierte Liste der Dateinamen mit Pfaden)**Beispiel:**

```
Dim pSession
```

```
Set pSession = CreateObject("axcdl.DrtSession")  
DrtObject.SaveFiles(pSession, 0, "C:\meinBild.tif")
```

Siehe: [Allgemeine Beispiele](#)

HRESULT GetFilename ([in] long *lIndex*, [out, retval] BSTR * *pbstrFilename*)

Beschreibung Gibt einen Dateinamen des Objektes aus der Liste (nur initialisiert nach einem Aufruf von LoadFiles).

Parameter:

LONG *lIndex* Index in der Dateiliste

Rückgabewerte:

*BSTR***bsFilename* Dateinamen inkl. Pfadangabe

Beispiel:

```
bsFilename = DrtObject.GetFileName(0)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT AddFilename ([in] BSTR *bstrFilename*, [out, retval] long * *plSize*)

Beschreibung Fügt einen Dateinamen an die Liste der Dateien des Objektes an).

Parameter:

BSTR *bsFilename* Dateiname inkl. Pfadangabe

Rückgabewerte:

*LONG***plSize* aktuelle Größe der Dateiliste

Beispiel:

```
DrtObject.AddFilename("C:\meinBild.tif")
```

Siehe: [Allgemeine Beispiele](#)

HRESULT GetFileNames ([out, retval] BSTR * *pbstrFiles*)

Beschreibung Gibt eine Liste der Dateinamen des Objektes zurück (nur initialisiert nach einem Aufruf von LoadFiles).

Rückgabewerte:

*BSTR***bsFileNames* Dateinamen (Komma separierte langer der Dateinamen mit Pfaden)

Beispiel:

```
bsFileNames = DrtObject.GetFileNames()
```

Siehe: [Allgemeine Beispiele](#)

HRESULT SetFileNames ([in] BSTR *bstrFiles*, [out, retval] long * *plSize*)

Beschreibung Setzt die Dateien des Objektes aus einer Liste.

Parameter:

BSTR bsFileNames Dateinamen (Komma separierte Liste der Dateinamen mit Pfaden)

Rückgabewerte:

*LONG** plIndex aktuelle Länge der Dateiliste

Beispiel:

```
DrtObject.SetFileNames(bsFileNames)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT InsertFilename ([in] long lIndex, [in] *BSTR* bstrFilename, [out, retval] long * plIndex)

Beschreibung Schreibt einen Dateinamen an eine bestimmte Stelle in der Dateiliste.

Parameter:

BSTR bsFilename Dateiname inkl. Pfadangabe

Rückgabewerte:

*LONG** plIndex aktuelle Index an den geschrieben wurde

Beispiel:

```
DrtObject.InsertFilename(2, "C:\meinBild.tif")
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Childs ([out, retval] [IDrtObjects](#) ** ppObjects)

Beschreibung Ermittelt die Kinderobjekte des aktuellen Objektes, wenn eine hierarchische Anfrage gestellt wurde.

Rückgabewerte:

*IDrtObjects*** ppObjects Liste mit Objekten, die unter dem aktuellen Objekt liegen.

Beispiel:

```
Childobjects = DrtObject.Childs
```

Siehe: [Allgemeine Beispiele](#)

HRESULT UndoCheckout ([in] [IDrtSession](#) *)

Beschreibung Wenn eine Datei über LoadFiles ausgecheckt wurde, kann dies mit dieser Funktion wieder rückgängig gemacht werden.

Beispiel:

```
DrtObject.UndoCheckout()
```

Siehe: [Allgemeine Beispiele](#)

HRESULT FileCount ([out, retval] long * plCount)

Beschreibung Diese Methode gibt die Anzahl der Dokumentdateien zurück

Rückgabewerte:

*LONG** plCount Anzahl des Objektdateien

Beispiel:

```
plCnt = DrtObject.FileCount
```

Siehe: [Allgemeine Beispiele](#)

HRESULT FileSize ([out, retval] long * *pSize*)

Beschreibung Diese Methode gibt die Größe der Dokumentdateien zurück

Rückgabewerte:

*LONG** pSize Größe des Objektdatei

Beispiel:

```
pSize = DrtObject.FileSize
```

Siehe: [Allgemeine Beispiele](#)

HRESULT FileExt ([out, retval] BSTR * *pbstrExt*)

Beschreibung Gibt die Dateierweiterung der Dateien zum Objekttyp an).

Rückgabewerte:

*BSTR** bsExt String mit Dateierweiterung

Beispiel:

```
bsExt = DrtObject.GetFileExt()
```

Siehe: [Allgemeine Beispiele](#)

HRESULT FileMimeType ([out, retval] BSTR * *pbstrExt*)

Beschreibung Gibt den Mimetype der Dateien zum Objekttyp an).

Rückgabewerte:

*BSTR** bsMime String mit Mimetypen

Beispiel:

```
bsMime = DrtObject.GetFileMimeType()
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Restore ([in] [IDrtSession](#) * *pSession*)

Beschreibung: Diese Methode stellt das vormalig gelöschte Objekt wieder her.

Parameter:

*IDrtSession** pSession Zeiger auf das IDrtSession-Objekt das die Verbindung zum Server hält

Beispiel:

```
Dim pSession  
Dim pObject  
  
Set pSession = CreateObject("axcdl.DrtSession")  
Set pObject = FindDeletedObjectFromSomewhere()
```

```
pObject.Restore(pSession)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT IsDeleted ([out, retval] long * *plIsDeleted*)

Beschreibung Gibt an, ob das Object im Papierkorb liegt.

Rückgabewerte:

*LONG*Flag, das angibt, ob das Objekt im Papierkorb liegt.

Beispiel:

```
isDeleted = DrtObject.IsDeleted
```

Siehe: [Allgemeine Beispiele](#)

IDrtObjects Schnittstellenreferenz

Abgeleitet von IDispatch.

Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Zusammenstellung von Objekten vom Typ [IDrtObject](#).

Siehe: [IDrtFields](#), [IDrtField](#), IDrtMultiFields, IDrtTableFields

Öffentliche Methoden

HRESULT [Deserialize](#) ([in] BSTR bstrXml)

HRESULT [Serialize](#) ([out, retval] BSTR *pbstrXml)

Dokumentation der Elementfunktionen

HRESULT Deserialize ([in] BSTR *bstrXml*)

Beschreibung: Diese Methode initialisiert ein [IDrtObjects](#) - Objekt einschliesslich seiner Attribute und Unterobjekte aus einem XML String.

Parameter:

BSTR bstrXml: String der das XML enthält, aus dem das Objekt initialisiert werden soll.

Beispiel:

```
DrtObjects.Deserialize(bstrXml)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Serialize ([out, retval] BSTR **pbstrXml*)

Beschreibung: Diese Methode serialisiert das [IDrtObjects](#) - Objekt einschliesslich seiner Attribute und Unterobjekte vollständig in einen XML String.

Rückgabewerte:

*BSTR** pbstrXml: String, der die XML Entsprechung des Objektes enthält

Beispiel:

```
bstrXml = DrtObjects.Serialize
```

Siehe: [Allgemeine Beispiele](#)

IDrtObjectTypeConditionNode Schnittstellenreferenz

Abgeleitet von IDispatch.

Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Bedingung auf einen Objekttyp, die zur Recherche nach DMS - Objekten verwendet werden kann.

Öffentliche Methoden

HRESULT [Deserialize](#) ([in] BSTR bstrXml)
 HRESULT [Serialize](#) ([out, retval] BSTR *pbstrXml)
 HRESULT [NodeType](#) ([out, retval] DrtConditionNodeTypeEnum *pNodeType)
 HRESULT [NodeType](#) ([in] DrtConditionNodeTypeEnum NodeType)
 HRESULT [ObjectType](#) ([out, retval] [IDrtObject](#) **ppObject)
 HRESULT [ObjectType](#) ([in] [IDrtObject](#) *pObject)
 HRESULT [FieldConditions](#) ([out, retval] [IDrtFieldConditionNode](#) **ppNode)
 HRESULT [FieldConditions](#) ([in] [IDrtFieldConditionNode](#) *pNode)

Dokumentation der Elementfunktionen

HRESULT Deserialize ([in] BSTR *bstrXml*)

Beschreibung: Diese Methode initialisiert ein [IDrtObjectTypeConditionNode](#) - Objekt einschließlich seiner Attribute und Unterobjekte aus einem XML String.

Parameter:

BSTR bstrXml String der das XML enthält, aus dem das Anfrageobjekt initialisiert werden soll.

Beispiel:

```
DrtConds.Deserialize(bstrXml)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Serialize ([out, retval] BSTR **pbstrXml*)

Beschreibung: Diese Methode serialisiert das [IDrtObjectTypeConditionNode](#) - Objekt einschließlich seiner Attribute und Unterobjekte vollständig in einen XML String.

Rückgabewerte:

*BSTR*pbstrXml* String, der die XML Entsprechung der Anfragebedingungen enthält

Beispiel:

```
bstrXml = DrtConds.Serialize
```

Siehe: [Allgemeine Beispiele](#)

HRESULT NodeType ([out, retval] DrtConditionNodeTypeEnum **pNodeType*)

Beschreibung: Gibt den Typ dieses Bedingungsobjektes zurück. Es werden hier die Typen DrtConditionNodeType_Leaf = 0 DrtConditionNodeType_Flat = 3 unterstützt. Ein Knoten vom Typ Flat enthält auf eine Liste von Bedingungsknoten vom Typ Leaf. Dieser Knoten vom Typ Flat enthält direkt keinen

Objektyp und keine Feldbedingungen. Er dient als Liste der Bedingungsknoten vom Typ Leaf. Ein Knoten vom Typ Leaf enthält die Angabe eines Objekttyps sowie eine Liste der zugeordneten Feldbedingungen.

Rückgabewerte:

*IDrtObjectTypeConditionNode** pCond Objekt, das eine Bedingung für die Anfrage enthält

Beispiel:

```
DrtConditionNodeTypeEnum*pNodeType = DrtObjectTypeConditionNode.NodeType
```

Siehe: [Allgemeine Beispiele](#)

HRESULT NodeType ([in] DrtConditionNodeTypeEnum *NodeType*)

Beschreibung: Gibt den Typ dieses Bedingungsobjektes zurück. Es werden hier die Typen DrtConditionNodeType_Leaf = 0 DrtConditionNodeType_Flat = 3 unterstützt. Ein Knoten vom Typ Flat enthält auf eine Liste von Bedingungsknoten vom Typ Leaf. Dieser Knoten vom Typ Flat enthält direkt keinen Objekttyp und keine Feldbedingungen. Er dient als Liste der Bedingungsknoten vom Typ Leaf. Ein Knoten vom Typ Leaf enthält die Angabe eines Objekttyps sowie eine Liste der zugeordneten Feldbedingungen.

Rückgabewerte:

*IDrtObjectTypeConditionNode** pCond Objekt, das eine Bedingung für die Anfrage enthält

Beispiel:

```
DrtObjectTypeConditionNode.NodeType = DrtConditionNodeType_Leaf
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ObjectType ([out, retval] [IDrtObject](#) ** *ppObject*)

Beschreibung: Diese Methode gibt den Objekttyp für diese Anfrage zurück. Der Objekttyp wird durch ein IDrtObject festgelegt. Enthält diese Objekt Feldeinträge ([IDrtFields](#)) so werden diese Felder angefragt (Evtl. eingetragene Feldwerte werden dabei ignoriert). Sind keine Felder vorhanden werden defaultmäßig alle Felder des Objektes angefragt. (Eigenschaft ist nicht wirksam wenn Knotentyp FLAT gewählt ist)

Rückgabewerte:

*IDrtObject** pObjectType Objekt, das den angefragten Objekttyp beschreibt

Beispiel:

```
DrtRequestObjectType = DrtQuery.ObjectType
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ObjectType ([in] [IDrtObject](#) * *pObject*)

Beschreibung: Diese Methode legt den Objekttyp für diese Anfrage fest. Der Objekttyp wird durch ein IDrtObject festgelegt. Enthält diese Objekt Feldeinträge ([IDrtFields](#)) so werden diese Felder angefragt (Evtl. eingetragene Feldwerte werden dabei ignoriert). Sind keine Felder vorhanden werden defaultmäßig alle Felder des Objektes angefragt. (Eigenschaft ist nicht wirksam wenn Knotentyp FLAT gewählt ist)

Rückgabewerte:

*IDrtObject** pObjectType Objekt, das den angefragten Objekttyp beschreibt

Beispiel:

```
DrtQuery.ObjectType = DrtRequestObjectType
```

Siehe: [Allgemeine Beispiele](#)

HRESULT FieldConditions ([out, retval] [IDrtFieldConditionNode](#) ** ppNode)

Beschreibung: Diese Methode gibt die Feldbedingungen zu der Anfrage eines Objekttyps zurück. Diese Feldbedingungen werden von einer baumartigen Bedingungsliste gebildet. Von jedem Objekt dieser Feldbedingungen kann auf das nächste Objekt zugegriffen werden. (Eigenschaft ist nicht wirksam wenn Knotentyp FLAT gewählt ist)

Rückgabewerte:

*IDrtFieldConditionNode** pFieldCond Objekt, das eine alle Feldbedingungen für die Anfrage zu einem Objekttyp enthält

Beispiel:

```
DrtRequestFieldConds = DrtConds.FieldConditions
```

Siehe: [Allgemeine Beispiele](#)

HRESULT FieldConditions ([in] [IDrtFieldConditionNode](#) * pNode)

Beschreibung: Diese Methode legt die Feldbedingungen zu der Anfrage eines Objekttyps fest. Diese Feldbedingungen werden von einer baumartigen Bedingungsliste gebildet. Von jedem Objekt dieser Feldbedingungen kann auf das nächste Objekt zugegriffen werden. (Eigenschaft ist nicht wirksam wenn Knotentyp FLAT gewählt ist)

Rückgabewerte:

*IDrtFieldConditionNode** pFieldCond Objekt, das eine alle Feldbedingungen für die Anfrage zu einem Objekttyp enthält

Beispiel:

```
DrtConds.FieldConditions = DrtRequestFieldConds
```

Siehe: [Allgemeine Beispiele](#)

IDrtPortfolio Schnittstellenreferenz

Abgeleitet von IDispatch.

Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Mappe mit den Objekte. Die Eigenschaften der Objekte entsprechen den gespeicherten Werten in der Datenbank- tabelle. Die Objekteigenschaften stellen die Mappeninhalte dar nach denen angelegt bzw. gesucht werden kann. Die Objektbezüge entsprechen derer die in der Datenbanktabelle gespeichert werden.

Öffentliche Methoden

HRESULT [Deserialize](#) ([in] BSTR bsXml)
 HRESULT [Serialize](#) ([out, retval] BSTR *pXml)
 HRESULT [Objects](#) ([out, retval] [IDrtObjects](#) **pVal)
 HRESULT [ID](#) ([out, retval] long *pVal)
 HRESULT [ID](#) ([in] long newVal)
 HRESULT [Created](#) ([out, retval] long *pVal)
 HRESULT [Created](#) ([in] long newVal)
 HRESULT [Creator](#) ([out, retval] BSTR *pVal)
 HRESULT [Creator](#) ([in] BSTR newVal)
 HRESULT [Creator_id](#) ([out, retval] long *pVal)
 HRESULT [Creator_id](#) ([in] long newVal)
 HRESULT [Recipient](#) ([out, retval] BSTR *pVal)
 HRESULT [Recipient](#) ([in] BSTR newVal)
 HRESULT [Recipient_id](#) ([out, retval] long *pVal)
 HRESULT [Recipient_id](#) ([in] long newVal)
 HRESULT [Subject](#) ([out, retval] BSTR *pVal)
 HRESULT [Subject](#) ([in] BSTR newVal)
 HRESULT [Public](#) ([out, retval] long *pVal)
 HRESULT [Public](#) ([in] long newVal)

Dokumentation der Elementfunktionen

HRESULT Deserialize ([in] BSTR *bsXml*)

Beschreibung: Diese Methode füllt die Mappen Felder mit Hilfe eines XML formatierten Strings.

Parameter:

BSTR bstrXml. XML Eingabestring

Beispiel:

```
DrtPortfolio.Deserialize(BSTR bstrXml)
```

Siehe: [Allgemeine Beispiele](#) [IDrtPortfolio](#)

HRESULT Serialize ([out, retval] BSTR **pXml*)

Beschreibung: Diese Methode konvertiert die Mappen Liste zu einem XML - Tag - formatierten String.

Rückgabewerte:

*BSTR**pXml: Ausgabe XML - String.

Beispiel:

```
BSTR pXml = DrtPortfolio.Serialize
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Objects ([out, retval] [IDrtObjects](#) ** pVal)

Beschreibung: Die Eigenschaft Objects definiert die Schnittstelle [IDrtObjects](#) zu einer ComCollection von DrtObject.

Die Eigenschaft Objects stellt die Mappeninhaltsliste von Objektbezügen dar.

Rückgabewerte:

*IDrtObjects*** pVal

Beispiel:

```
IDrtObjects**pVal = DrtPortfolio.Objects
```

Siehe: [Allgemeine Beispiele](#)

Bemerkungen:

Diese Eigenschaft kann gelesen werden.

HRESULT ID ([out, retval] long * pVal)

Beschreibung: Die Eigenschaften ID definiert das Mappen ID Feld.

Rückgabewerte:

*LONG** pVal

Beispiel:

```
LONG* pVal = DrtPortfolio.ID
```

Siehe: [Allgemeine Beispiele](#)

Bemerkungen:

Diese Eigenschaft kann gelesen und geschrieben werden.

HRESULT ID ([in] long newVal)

Beschreibung: Die Eigenschaften ID definiert das Mappen ID Feld.

Parameter:

LONG newVal

Beispiel:

```
DrtPortfolio.ID (LONG newVal)
```

Siehe: [Allgemeine Beispiele](#)

Bemerkungen:

Diese Eigenschaft kann gelesen und geschrieben werden.

HRESULT Created ([out, retval] long * *pVal*)

Beschreibung: Die Eigenschaft Created definiert das Mappen Datumsfeld. Gültiges Datum und Uhrzeit wäre minimal Januar 1. 1970, 00:00:00.

Rückgabewerte:

*LONG**pVal

Beispiel:

```
LONG* pVal = DrtPortfolio.Created
```

Siehe: [Allgemeine Beispiele](#)

Bemerkungen:

Diese Eigenschaft kann gelesen und geschrieben werden.

HRESULT Created ([in] long *newVal*)

Beschreibung: Die Eigenschaft Created definiert das Mappen Datumsfeld. Gültiges Datum und Uhrzeit wäre minimal Januar 1. 1970, 00:00:00.

Parameter:

*LONG*newVal

Beispiel:

```
DrtPortfolio.Created (LONG newVal)
```

Siehe: [Allgemeine Beispiele](#)

Bemerkungen:

Diese Eigenschaft kann gelesen und geschrieben werden.

HRESULT Creator ([out, retval] BSTR * *pVal*)

Beschreibung: Die Eigenschaft Creator definiert das Mappen Erstellerfeld (Benutzer).

Rückgabewerte:

*BSTR**pVal

Beispiel:

```
BSTR* pVal = DrtPortfolio.Creator
```

Siehe: [Allgemeine Beispiele](#)

Bemerkungen:

Diese Eigenschaft kann gelesen und geschrieben werden.

HRESULT Creator ([in] BSTR *newVal*)

Beschreibung: Die Eigenschaft Creator definiert das Mappen Erstellerfeld (Benutzer).

Parameter:*BSTR newVal***Beispiel:**

```
DrtPortfolio.Creator (BSTR newVal)
```

Siehe: [Allgemeine Beispiele](#)

Bemerkungen:

Diese Eigenschaft kann gelesen und geschrieben werden.

HRESULT Creator_id ([out, retval] long * *pVal*)

Beschreibung: Die Eigenschaft `Creator_id` definiert das Mappen Erstelleridfeld.

Rückgabewerte:*LONG* pVal***Beispiel:**

```
LONG* pVal = DrtPortfolio.Creator_id
```

Siehe: [Allgemeine Beispiele](#)

Bemerkungen:

Diese Eigenschaft kann gelesen und geschrieben werden.

HRESULT Creator_id ([in] long *newVal*)

Beschreibung: Die Eigenschaft `Creator_id` definiert das Mappen Erstelleridfeld.

Parameter:*LONG newVal***Beispiel:**

```
DrtPortfolio.Creator_id (LONG pVal)
```

Siehe: [Allgemeine Beispiele](#)

Bemerkungen:

Diese Eigenschaft kann gelesen und geschrieben werden.

HRESULT Recipient ([out, retval] BSTR * *pVal*)

Beschreibung: Die Eigenschaft `Recipient` definiert das Mappen Empfängerfeld.

Rückgabewerte:*BSTR* pVal***Beispiel:**

```
BSTR* pVal = DrtPortfolio.Recipient
```

Siehe: [Allgemeine Beispiele](#)

Bemerkungen:

Diese Eigenschaft kann gelesen und geschrieben werden.

HRESULT Recipient ([in] BSTR *newVal*)

Beschreibung: Die Eigenschaft Recipient definiert das Mappen Empfängerfeld.

Parameter:

BSTR newVal

Beispiel:

```
DrtPortfolio.Recipient(newVal)
```

Siehe: [Allgemeine Beispiele](#)

Bemerkungen:

Diese Eigenschaft kann gelesen und geschrieben werden.

HRESULT Recipient_id ([out, retval] long **pVal*)

Beschreibung: Die Eigenschaft Recipient_id definiert das Mappen Empfängeridfeld. Es erhält die ID vom Empfänger.

Rückgabewerte:

*LONG** pVal

Beispiel:

```
LONG* pVal = DrtPortfolio.Recipient_id
```

Siehe: [Allgemeine Beispiele](#)

Bemerkungen:

Diese Eigenschaft kann gelesen und geschrieben werden.

HRESULT Recipient_id ([in] long *newVal*)

Beschreibung: Die Eigenschaft Recipient_id definiert das Mappen Empfängeridfeld. Es schreibt die ID vom Empfänger in die entsprechende Datenbanktabelle.

Parameter:

LONG newVal

Beispiel:

```
DrtPortfolio.Recipient_id(LONG newVal)
```

Siehe: [Allgemeine Beispiele](#)

Bemerkungen:

Diese Eigenschaft kann gelesen und geschrieben werden.

HRESULT Subject ([out, retval] BSTR * *pVal*)

Beschreibung: Die Eigenschaft Subject definiert das Feld "zum Thema" (Betreff) auf der Mappe.

Rückgabewerte:

*BSTR** *pVal*

Beispiel:

```
BSTR* pVal = DrtPortfolio.Subject
```

Siehe: [Allgemeine Beispiele](#)

Bemerkungen:

Diese Eigenschaft kann gelesen und geschrieben werden.

HRESULT Subject ([in] BSTR *newVal*)

Beschreibung: Die Eigenschaft Subject definiert das Feld "zum Thema" (Betreff) auf der Mappe.

Parameter:

BSTR *newVal*

Beispiel:

```
DrtPortfolio.Subject(BSTR newVal)
```

Siehe: [Allgemeine Beispiele](#)

Bemerkungen:

Diese Eigenschaft kann gelesen und geschrieben werden.

HRESULT Public ([out, retval] long * *pVal*)

Beschreibung: Eigenschaften definiert das Public - Flag eines Portfolios

Rückgabewerte:

*LONG** *pVal*

Bemerkungen:

Diese Eigenschaft kann gelesen und geschrieben werden.

HRESULT Public ([in] long *newVal*)

Beschreibung: Eigenschaften definiert das Public - Flag eines Portfolios

Parameter:

long *newVal*

Bemerkungen:

Diese Eigenschaft kann gelesen und geschrieben werden.

IDrtPortfolioMgr Schnittstellenreferenz

Abgeleitet von IDispatch.

Ausführliche Beschreibung

Diese Schnittstelle ermöglicht den Zugriff auf Mappen (Portfolios). Es können Mappen angelegt, bearbeitet, gelöscht und recherchiert werden.

Öffentliche Methoden

HRESULT [AddPortfolio](#) ([in] BSTR Subject,[in] BSTR Recipient,[in] long bPublic,[in] [IDrtObjects](#) *pObjects,[in] long Mode,[out, retval] long *pPortfolioID)
 HRESULT [ModPortfolio](#) ([in] long PortfolioID,[in] BSTR Subject,[in] BSTR Recipient,[in] [IDrtObjects](#) *pObjects,[in] long Mode,[out, retval] long *pResult)
 HRESULT [DelPortfolio](#) ([in] long PortfolioID,[in] BSTR Subject,[in] BSTR Recipient,[out, retval] long *pResult)
 HRESULT [RetrievePortfolios](#) ([in] long PortfolioID,[in] BSTR Subject,[in] BSTR Recipient,[in] long Created_from,[in] long Created_to,[in] BSTR Creator,[out, retval] [IDrtPortfolios](#) **ppPotrfolios)

Dokumentation der Elementfunktionen

HRESULT AddPortfolio ([in] BSTR *Subject*, [in] BSTR *Recipient*, [in] long *bPublic*, [in] [IDrtObjects](#) **pObjects*, [in] long *Mode*, [out, retval] long **pPortfolioID*)

Beschreibung: Diese Methode erstellt eine neue Mappe und fügt Objektbezüge zu den vorhandenen Objekten oder Mappen hinzu. Diese Methode ist eine Kapselung für dms.AddPortfolio Job.

Parameter:

BSTR Subject: Subjekt der Mappe. Darf nicht leer sein.

BSTR Recipient: Empfänger der Mappe. Darf nicht leer sein.

IDrtObjects pObjects*: Abbild von [IDrtObjects](#) mit der Zusammenstellung von Objekten, zum hinzufügen zur Mappe.

LONG Mode:

0 - hinzufügen von neuen Objekten nur in existierenden Mappen

1 - erstellt eine Mappe.

Rückgabewerte:

*LONG*PortfolioID*: ID der angelegten oder geänderten Mappe.

Hinweis: mögliche Fehlercodes

-1 Fehler

Beispiel:

```
Dim pObjects

Set pObjects = CreateObject("axcdl.Objects")

PortfolioID = DrtPortfolioMgr.AddPortfolio(Subject, Recipient, pObjects,
Mode)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ModPortfolio ([in] long *PortfolioID*, [in] BSTR *Subject*, [in] BSTR *Recipient*, [in] [IDrtObjects](#) * *pObjects*, [in] long *Mode*, [out, retval] long * *pResult*)

Beschreibung: Diese Methode fügt Objektbezüge zu existierenden Mappen hinzu. Diese Methode ist eine Kapselung für dms.ModPortfolio Job.

Parameter:

*LONG*PortfolioID: ID für das Updaten der Mappe. Darf nie gleich -1 sein. Subjekt und Empfänger dürfen nicht umhüllt sein.

BSTR Subject: Subjekt der Mappe. Darf nicht leer sein wenn PortfolioID = -1.

BSTR Recipient: Empfänger der Mappe. Darf nicht leer sein wenn PortfolioID = -1.

*IDrtObjects** *pObjects*: Abbild vom [IDrtObjects](#) mit der Zusammenstellung von Objekten, zum hinzufügen von Mappen.

*LONG*Mode:

0 - hinzufügen eines neuen Objektes zur Portfolio

1 - löschen eines alten Inhalts vom Portfolio und hinzufügen eines neuen Inhalts vom der Objektzusammenstellung.

Rückgabewerte:

*LONG** *pResult*

Hinweis: mögliche Fehlercodes

0 - kein Fehler

oder dms.ModPortfolio Job hat als Ergebnis einen Fehler (Servercode)

Beispiel:

```
Dim pObjects

Set pObjects = CreateObject("axcdl.Objects")

pResult = DrtPortfolioMgr.ModPortfolio(PortfolioID, Subject, Recipient,
pObjects, Mode)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT DelPortfolio ([in] long *PortfolioID*, [in] BSTR *Subject*, [in] BSTR *Recipient*, [out, retval] long * *pResult*)

Beschreibung: Diese Methode löscht Mappen und deren Inhalt. Diese Methode ist eine Kapselung für den dms.DelPortfolio Job.

Zur Identifizierung der Mappe reicht es aus die ID anzugeben oder Recipient und Subject. Somit kann die Parametereingabe im Beispiel eingränzen.

Parameter:

*LONG*PortfolioID: ID von der zu löschenden Mappe. Darf nie gleich -1 sein. Subjekt und Empfänger dürfen nicht umhüllt sein.

BSTR Subject: Subjekt der Mappe. Darf nicht leer sein wenn PortfolioID = -1.

BSTR Recipient: Empfänger der Mappe. Darf nicht leer sein wenn PortfolioID = -1.

Rückgabewerte:

*LONG** *pResult*

Hinweis: mögliche Fehlercodes

0 - kein Fehler

oder dms.DelPortfolio Job hat als Ergebnis einen Fehler (Servercode)

Beispiel:

```
pResult = DrtPortfolioMgr.DelPortfolio(PortfolioID, Subject, Recipient)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT RetrievePortfolios ([in] long *PortfolioID*, [in] BSTR *Subject*, [in] BSTR *Recipient*, [in] long *Created_from*, [in] long *Created_to*, [in] BSTR *Creator*, [out, retval] [IDrtPortfolios](#) ** *ppPotrfolios*)

Beschreibung: Diese Methode gibt eine Liste mit Mappen ([IDrtPortfolio](#)) zurück die den angefragten Suchbedingungen genügen.

Parameter:

*LONG*PortfolioID - ID der abzurufenden Mappe
BSTR Subject: Betreff der Mappe
BSTR Recipient: Empfänger der Mappe
BSTR UserName: Ersteller der Mappe
*LONG*Created_from: niedrigste Grenze der Erstellzeit
*LONG*Created_to: höchste Grenze der Erstellzeit

Ist die PortfolioID = -1, so werden die Parameter Betreff, Empfänger, Erstellzeit von, Erstellzeit bis und Ersteller für die Anfrage benutzt.

Werden die Felder Betreff, Empfänger und Ersteller mit einen Leerstring gefüllt so werden Diese nicht in der Anfrage berücksichtigt.

Ist die Erstellzeit = 0 oder "" so wird auch Diese nicht in der Anfrage mit berücksichtigt.

Rückgabewerte:

*IDrtPortfolios*** *ppPortfolios*: Liste der gefundenen Mappen

Beispiel:

```
// Beispiel für "alle Mappen abrufen"
ppPortfolios = IDrtPortfolioMgr.RetrievePortfolios(-1, '', '', 0, 0, '')

// Beispiel für "alle abrufen die, erstellt wurden von benutzer2 und der
Empfänger benutzer1 war"
ppPortfolios = IDrtPortfolioMgr.RetrievePortfolios(-
1, '', 'benutzer1', 0, 0, 'benutzer2')

// Beispiel allgemeine Zeile
ppPortfolios = IDrtPortfolioMgr.RetrievePortfolios(PortfolioID, Subject,
Recipient, Created_from, Created_to, UserName)
```

Siehe: [Allgemeine Beispiele](#)

IDrtPortfolios Schnittstellenreferenz

Abgeleitet von IDispatch.

Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Mappe mit den Objekte. Die Eigenschaften der Objekte entsprechen den gespeicherten Werten in der Datenbank- tabelle. Die Objekteigenschaften stellen die Mappeninhalte dar nach denen angelegt bzw. gesucht werden kann.Die Objektbezüge entsprechen derer die in der Datenbanktabelle gespeichert werden.

Siehe auch: [IDrtPortfolio](#)

Öffentliche Methoden

HRESULT [Deserialize](#) ([in] BSTR bstrXml)

HRESULT [Serialize](#) ([out, retval] BSTR *pXml)

Dokumentation der Elementfunktionen

HRESULT Deserialize ([in] BSTR *bstrXml*)

Beschreibung: Diese Methode füllt die Mappen Felder durch einen XML String.

Parameter:

BSTR bstrXml XML Eingabestring

Beispiel:

```
DrtPortfolios.Deserialize (BSTR bstrXml)
```

Siehe: [Allgemeine Beispiele](#) [IDrtPortfolio](#)

HRESULT Serialize ([out, retval] BSTR **pXml*)

Beschreibung: Diese Methode konvertiert die Mappen Liste zu einem XML - Tag - formatierten String.

Rückgabewerte:

*BSTR** pXml: Ausgabe XML - String

Beispiel:

```
BSTR* pXml = DrtPortfolios.Serialize
```

Siehe: [Allgemeine Beispiele](#) [IDrtPortfolio](#)

IDrtQueries Schnittstellenreferenz

Abgeleitet von IDispatch.

Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Zusammenstellung von Objekten vom Typ [IDrtQuery](#). Zusätzlich kann die maximale Größe der Trefferliste beschränkt werden, bzw. Einstellungen zum seitenweisen Auslesen der Trefferliste vorgenommen werden.

Siehe: [IDrtQuery](#)

Öffentliche Methoden

HRESULT [Deserialize](#) ([in] BSTR bstrXml)
HRESULT [Serialize](#) ([out, retval] BSTR *pbstrXml)
HRESULT [Name](#) ([out, retval] BSTR *pbstrName)
HRESULT [MaxHits](#) ([out, retval] long *pVal)
HRESULT [MaxHits](#) ([in] long newVal)
HRESULT [Offset](#) ([out, retval] long *pVal)
HRESULT [Offset](#) ([in] long newVal)
HRESULT [PageSize](#) ([out, retval] long *pVal)
HRESULT [PageSize](#) ([in] long newVal)
HRESULT [Flags](#) ([out, retval] long *pVal)
HRESULT [Flags](#) ([in] long newVal)

Dokumentation der Elementfunktionen

HRESULT Deserialize ([in] BSTR *bstrXml*)

Beschreibung: Diese Methode initialisiert ein [IDrtQueries](#) - Objekt einschliesslich seiner Attribute und Unterobjekte aus einem XML String.

Parameter:

BSTR bstrXml String der das XML enthält, aus dem die Liste mit Anfrageobjekten initialisiert werden soll.

Beispiel:

```
DrtQueries.Deserialize(bstrXml)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Serialize ([out, retval] BSTR **pbstrXml*)

Beschreibung: Diese Methode serialisiert das [IDrtQueries](#) - Objekt einschliesslich seiner Attribute und Unterobjekte vollständig in einen XML String.

Rückgabewerte:

*BSTR**pbstrXml String, der die XML Entsprechung die Liste Anfrageobjekte enthält

Beispiel:

```
bstrXml = DrtQueries.Serialize
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Name ([out, retval] BSTR * *pbstrName*)

Beschreibung: Diese Methode gibt den Namen der Anfrage an. Diese Eigenschaft ist nur initialisiert, wenn das Objekt vom Server als gespeicherte Anfrage ausgelesen wird. .

Rückgabewerte:

*BSTR***pbstrName* String, der den Namen der gespeicherten Anfrage enthält

Beispiel:

```
bstrName = DrtQueries.Name
```

Siehe: [Allgemeine Beispiele](#)

HRESULT MaxHits ([out, retval] long * *pVal*)

Beschreibung: Mit Hilfe dieser Eigenschaft kann die Anzahl der Ergebnisse für eine Anfrage beschränkt werden. Wird diese Eigenschaft gesetzt kann es dazu führen das Objekte bei einer Anfrage nicht gefunden werden.

Rückgabewerte:

*LONG***pVal* Wert, der die Maximale Trefferanzahl enthält

Beispiel:

```
MaximumHits = DrtQueries.MaxHits
```

Siehe: [Allgemeine Beispiele](#)

HRESULT MaxHits ([in] long *newVal*)

Beschreibung: Mit Hilfe dieser Eigenschaft kann die Anzahl der Ergebnisse für eine Anfrage beschränkt werden. Wird diese Eigenschaft gesetzt kann es dazu führen das Objekte bei einer Anfrage nicht gefunden werden.

Parameter:

long *pVal* Wert, der die Maximale Trefferanzahl enthält

Beispiel:

```
DrtQueries.MaxHits = MaximumHits
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Offset ([out, retval] long * *pVal*)

Beschreibung: Es ist möglich bei Anfragen mit vielen Treffern diese Anfragen seitenweise auszuwerten. Dazu wird mit der dieser Eigenschaft die lfd Nummer des ersten Ergebnisses dieser Anfrageseite angegeben. Die Maximale Anzahl der Treffer je Seite wird mit der Eigenschaft *PageSize* begrenzt.

Rückgabewerte:

*LONG***pVal* Wert, der die Nummer des ersten Treffers angibt

Beispiel:

```
offset = DrtQueries.Offset
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Offset ([in] long *newVal*)

Beschreibung: Es ist möglich bei Anfragen mit vielen Treffern diese Anfragen seitenweise auszuwerten. Dazu wird mit der dieser Eigenschaft die lfd. Nummer des ersten Ergebnisses dieser Anfrageseite angegeben. Die Maximale Anzahl der Treffer je Seite wird mit der Eigenschaft *PageSize* begrenzt.

Parameter:

long *pVal* Wert, der die Nummer des ersten Treffers angibt

Beispiel:

```
DrtQueries.Offset = offset
```

Siehe: [Allgemeine Beispiele](#)

HRESULT PageSize ([out, retval] long * *pVal*)

Beschreibung: Es ist möglich bei Anfragen mit vielen Treffern diese Anfragen seitenweise auszuwerten. Dazu wird mit der dieser Eigenschaft die maximale Trefferanzahl dieser Ergebnisseite pro Objekttyp begrenzt. Das seitenweise auswerten der Ergebnisse erfolgt dann über Variation der Eigenschaft *Offset*, die jeweils die lfd. Nummer des ersten Ergebnisses der Seite angibt.

Rückgabewerte:

*LONG***pVal* Wert, der die Maximale Trefferanzahl pro Anfrageseite enthält

Beispiel:

```
MaximumHits = DrtQueries.MaxHits
```

Siehe: [Allgemeine Beispiele](#)

HRESULT PageSize ([in] long *newVal*)

Beschreibung: Es ist möglich bei Anfragen mit vielen Treffern diese Anfragen seitenweise auszuwerten. Dazu wird mit der dieser Eigenschaft die maximale Trefferanzahl dieser Ergebnisseite pro Objekttyp begrenzt. Das seitenweise auswerten der Ergebnisse erfolgt dann über Variation der Eigenschaft *Offset*, die jeweils die lfd. Nummer des ersten Ergebnisses der Seite angibt.

Parameter:

long *pVal* Wert, der die Maximale Trefferanzahl pro Anfrageseite enthält

Beispiel:

```
DrtQueries.MaxHits = MaximumHits
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Flags ([out, retval] long * *pVal*)

Beschreibung: Über verschiedene Flags kann hier eine Server Anfrage genauestens konfiguration werden.

Rückgabewerte:

*LONG***pVal* Wert, der die Flags zur Konfiguration der Anfrage enthält

Beispiel:

```
flags = DrtQueries.Flags
```

0 - Default

1 - Standard - Systemfelder werden mit abgefragt

Siehe: [Allgemeine Beispiele](#)

HRESULT Flags ([in] long *newVal*)

Beschreibung: Über verschiedene Flags kann hier eine Server Anfrage genauestens konfiguration werden.

Parameter:

long newVal Wert, der die Flags zur Konfiguration der Anfrage setzt

Beispiel:

```
DrtQueries.Flags = 0x1
```

0 - Default

1 - Standard - Systemfelder werden mit abgefragt

Siehe: [Allgemeine Beispiele](#)

IDrtQuery Schnittstellenreferenz

Abgeleitet von IDispatch.

Ausführliche Beschreibung

Mit diesem Objekt kann die Suche nach DMS - Objekten eingeschränkt werden. Es können Einschränkungen auf das Objekt selber oder auf Indexdatenfelder dieses Objektes gesetzt werden.

Siehe: IDrtConditions , IDrtFieldConditions

Öffentliche Methoden

HRESULT [Deserialize](#) ([in] BSTR bstrXml)
 HRESULT [Serialize](#) ([out, retval] BSTR *pbstrXml)
 HRESULT [Object](#) ([out, retval] [IDrtObject](#) **ppObject)
 HRESULT [Conditions](#) ([out, retval] [IDrtObjectTypeConditionNode](#) **ppNode)
 HRESULT [Conditions](#) ([in] [IDrtObjectTypeConditionNode](#) *pNode)
 HRESULT [Object](#) ([in] [IDrtObject](#) *pObject)
 HRESULT [ChildObjectsDepth](#) ([out, retval] long *plDepth)
 HRESULT [ChildObjectsDepth](#) ([in] long lDepth)

Dokumentation der Elementfunktionen

HRESULT Deserialize ([in] BSTR *bstrXml*)

Beschreibung: Diese Methode initialisiert ein [IDrtQuery](#) - Objekt einschliesslich seiner Attribute und Unterobjekte aus einem XML String.

Parameter:

BSTR bstrXml String der das XML enthält, aus dem das Anfrageobjekt initialisiert werden soll.

Beispiel:

```
DrtQuery.Deserialize(bstrXml)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Serialize ([out, retval] BSTR **pbstrXml*)

Beschreibung: Diese Methode serialisiert das [IDrtQuery](#) - Objekt einschliesslich seiner Attribute und Unterobjekte vollständig in einen XML String.

Rückgabewerte:

*BSTR**pbstrXml String, der die XML Entsprechung des einzelnen Anfrageobjekt enthält

Beispiel:

```
bstrXml = DrtQuery.Serialize
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Object ([out, retval] [IDrtObject](#) ** *ppObject*)

Beschreibung: Diese Methode gibt ein Anfrageobjekt an. Dieses Anfrageobjekt definiert den angefragten Objekttyp und die angefragten Felder.

Rückgabewerte:

*IDrtObject** pObject Objekt, das den angefragten Objekttyp enthält

Beispiel:

```
DrtQuery.Object = DrtObject
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Conditions ([out, retval] [IDrtObjectTypeConditionNode](#) ** ppNode)

Beschreibung: Diese Methode legt eine Bedingung dieser Anfrage fest. Diese Bedingungen enthalten Suchkriterien auf Objekt- oder auf Feldebene.

Rückgabewerte:

*IDrtObjectTypeConditionNode*** ppCond Objekt, das eine Bedingung für die Anfrage enthält

Beispiel:

```
DrtConds = DrtQuery.Conditions
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Conditions ([in] [IDrtObjectTypeConditionNode](#) * pNode)

Beschreibung: Diese Methode legt eine Bedingung dieser Anfrage fest. Diese Bedingungen enthalten Suchkriterien auf Objekt- oder auf Feldebene.

Rückgabewerte:

*IDrtObjectTypeConditionNode** pCond Objekt, das eine Bedingung für die Anfrage enthält

Beispiel:

```
DrtQuery.Conditions = DrtConds
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Object ([in] [IDrtObject](#) * pObject)

Beschreibung: Diese Methode legt fest, welcher Objekttyp bei der Recherche angefragt werden soll. Werden für den Objekttyp Felder festgelegt, werden nur diese Felder angefragt. Per Default werden alle Felder zurückgegeben.

Rückgabewerte:

*IDrtObject** pObject Objekt, den Objekttyp enthält, der bei der Suche zurückgegeben werden soll.

Beispiel:

```
DrtQuery.Objectct = DrtObject
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ChildObjectsDepth ([out, retval] long * pDepth)

Beschreibung: Es können Kindobjekte unterhalb des angefragten Objekttyps zurückgegeben werden. Diese Option ist nur bei Anfragen auf Ordner oder Register verfügbar. Dabei ist festlegbar, auf welchen Hierarchiestufen unterhalb des angefragten Objektes Kindobjekte ermittelt werden sollen.

Rückgabewerte:

*long*plDepth* Tiefe für Hierarchische Anfragen auf Ordner/Registern

Beispiel:

```
depth = DrtQuery.ChildObjectsDept
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ChildObjectsDepth ([in] long lDepth)

Beschreibung: Es können Kindobjekte unterhalb des angefragten Objekttypes zurückgegeben werden. Diese Option ist nur bei Anfragen auf Ordner oder Register verfügbar. Dabei ist festlegbar, auf welchen Hierarchiestufen unterhalb des angefragten Objektes Kindobjekte ermittelt werden sollen.

Parameter:

long lDepth Tiefe für Hierarchische Anfragen auf Ordner/Registern

Beispiel:

```
DrtQuery.ChildObjectsDept = 1
```

Siehe: [Allgemeine Beispiele](#)

IDrtRelation Schnittstellenreferenz

Abgeleitet von IDispatch.

Ausführliche Beschreibung

Diese Schnittstelle stellt die Relations Objekte zur Verfügung.

Die Eigenschaften von diesen Objekten werden aus eine Datenbanktabelle abgerufen.

Öffentliche Methoden

HRESULT [Deserialize](#) ([in] BSTR bstrXml)
 HRESULT [Serialize](#) ([out, retval] BSTR *pXml)
 HRESULT [Obj_id1](#) ([out, retval] long *pVal)
 HRESULT [Obj_id1](#) ([in] long newVal)
 HRESULT [Obj_typ1](#) ([out, retval] long *pVal)
 HRESULT [Obj_typ1](#) ([in] long newVal)
 HRESULT [Obj_id2](#) ([out, retval] long *pVal)
 HRESULT [Obj_id2](#) ([in] long newVal)
 HRESULT [Obj_typ2](#) ([out, retval] long *pVal)
 HRESULT [Obj_typ2](#) ([in] long newVal)
 HRESULT [Valid_from](#) ([out, retval] long *pVal)
 HRESULT [Valid_from](#) ([in] long newVal)
 HRESULT [Valid_to](#) ([out, retval] long *pVal)
 HRESULT [Valid_to](#) ([in] long newVal)
 HRESULT [Ensured](#) ([out, retval] long *pVal)
 HRESULT [Ensured](#) ([in] long newVal)
 HRESULT [Checked](#) ([out, retval] long *pVal)
 HRESULT [Checked](#) ([in] long newVal)
 HRESULT [RelID](#) ([out, retval] BSTR *pVal)
 HRESULT [RelID](#) ([in] BSTR newVal)
 HRESULT [RelTextID](#) ([out, retval] BSTR *pVal)
 HRESULT [RelTextID](#) ([in] BSTR newVal)
 HRESULT [IObjID](#) ([out, retval] long *pVal)
 HRESULT [IObjID](#) ([in] long newVal)
 HRESULT [IObjType](#) ([out, retval] long *pVal)
 HRESULT [IObjType](#) ([in] long newVal)
 HRESULT [Created](#) ([out, retval] long *pVal)
 HRESULT [Created](#) ([in] long newVal)
 HRESULT [CreatedBy](#) ([out, retval] BSTR *pVal)
 HRESULT [CreatedBy](#) ([in] BSTR newVal)
 HRESULT [ModifyTime](#) ([out, retval] long *pVal)
 HRESULT [ModifyTime](#) ([in] long newVal)
 HRESULT [ModifyUser](#) ([out, retval] BSTR *pVal)
 HRESULT [ModifyUser](#) ([in] BSTR newVal)
 HRESULT [Deleted](#) ([out, retval] long *pVal)
 HRESULT [DeleteUser](#) ([out, retval] BSTR *pVal)

Dokumentation der Elementfunktionen

HRESULT Deserialize ([in] BSTR *bstrXml*)

Beschreibung: Diese Methode setzt die Objekteigenschaften durch eine XML formatierten String.

Parameter:

BSTR bstrXml: xml-input string

Beispiel:

```
IDrtRelation.Deserialize(BSTR bstrXml)
```

Siehe: [Allgemeine Beispiele](#) , [Schnittstelle für den Zugriff und die Verwaltung von Relationen](#)

HRESULT Serialize ([out, retval] BSTR * *pXml*)

Beschreibung: Diese Methode erhält eine Objekteigenschaft in Form eines XML formatierten Strings.

Rückgabewerte:

*BSTR** *pXml*: output xml-string.

Beispiel:

```
BSTR* pXml = IDrtRelation.Serialize
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Obj_id1 ([out, retval] long * *pVal*)

Beschreibung: Die Eigenschaft definiert das object_id Feld vom from-object.

Rückgabewerte:

*LONG** *pVal*

Beispiel:

```
LONG* pVal = IDrtRelation.Obj_id1
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Obj_id1 ([in] long *newVal*)

Beschreibung: Die Eigenschaft definiert das object_id Feld vom from-object.

Parameter:

LONG *newVal*

Beispiel:

```
IDrtRelation.Obj_id1 (LONG newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Obj_typ1 ([out, retval] long * *pVal*)

Beschreibung: Die Eigenschaft definiert das object_type Feld vom from-object.

Rückgabewerte:

*LONG** *pVal*

Beispiel:

```
LONG* pVal = IDrtRelation.Obj_typ1
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Obj_typ1 ([in] long *newVal*)

Beschreibung: Die Eigenschaft definiert das object_type Feld vom from-object.

Parameter:

*LONG*newVal

Beispiel:

```
IDrtRelation.Obj_typ1 (LONG newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Obj_id2 ([out, retval] long **pVal*)

Beschreibung: Die Eigenschaft definiert das object_id Feld vom to-object.

Rückgabewerte:

*LONG**pVal

Beispiel:

```
LONG* pVal = IDrtRelation.Obj_id2
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Obj_id2 ([in] long *newVal*)

Beschreibung: Die Eigenschaft definiert das object_id Feld vom to-object.

Parameter:

*LONG*newVal

Beispiel:

```
IDrtRelation.Obj_id2 (LONG newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Obj_typ2 ([out, retval] long **pVal*)

Beschreibung: Die Eigenschaft definiert das object_type Feld vom to-object.

Rückgabewerte:

*LONG**pVal

Beispiel:

```
LONG* pVal = IDrtRelation.Obj_typ2
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Obj_typ2 ([in] long *newVal*)

Beschreibung: Die Eigenschaft definiert das object_type Feld vom to-object.

Parameter:

*LONG*newVal

Beispiel:

```
IDrtRelation.Obj_typ2(LONG newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Valid_from ([out, retval] long * pVal)

Beschreibung: Die Eigenschaft definiert die Beziehung des valid_from Feld. Die Sekunden der abgelaufene Zeit seit Mitternacht (00:00:00), 1. Januar, 1970.

Rückgabewerte:

*LONG** pVal

Beispiel:

```
LONG* pVal = IDrtRelation.Valid_from
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Valid_from ([in] long newVal)

Beschreibung: Die Eigenschaft definiert die Beziehung des valid_from Feld. Die Sekunden der abgelaufene Zeit seit Mitternacht (00:00:00), 1. Januar, 1970.

Parameter:

LONG newVal

Beispiel:

```
IDrtRelation.Valid_from(LONG newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Valid_to ([out, retval] long * pVal)

Beschreibung: Die Eigenschaft definiert die Beziehung des valid_to Feld. Die Sekunden der abgelaufene Zeit seit Mitternacht (00:00:00), 1. Januar, 1970.

Rückgabewerte:

*LONG** pVal

Beispiel:

```
LONG* pVal = IDrtRelation.Valid_to
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Valid_to ([in] long newVal)

Beschreibung: Die Eigenschaft definiert die Beziehung valid_to Feld. Die Sekunden der abgelaufene Zeit seit Mitternacht (00:00:00), 1. Januar, 1970.

Parameter:

LONG newVal

Beispiel:

```
IDrtRelation.Valid_to(LONG newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Ensured ([out, retval] long * pVal)

Beschreibung: Die Eigenschaft definiert die Beziehung zum Sicherstellungs - Feld. 0 - Nein, 1 - Ja

Rückgabewerte:

*LONG** pVal

Beispiel:

```
LONG* pVal = IDrtRelation.Ensured
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Ensured ([in] long newVal)

Beschreibung: Die Eigenschaft definiert die Beziehung zum Sicherstellungs - Feld. 0 - Nein, 1 - Ja

Parameter:

LONG newVal

Beispiel:

```
IDrtRelation.Ensured(LONG newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Checked ([out, retval] long * pVal)

Beschreibung: Die Eigenschaft definiert die Beziehung des Check Feldes. 0 - Nein, 1 - Ja

Rückgabewerte:

*LONG** pVal

Beispiel:

```
LONG* pVal = IDrtRelation.Checked
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Checked ([in] long newVal)

Beschreibung: Eigenschaft definiert die Beziehung des Check Feldes. 0 - Nein, 1 - Ja

Parameter:

LONG newVal

Beispiel:

```
IDrtRelation.Checked(LONG newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT RelID ([out, retval] BSTR * *pVal*)

Beschreibung: Die Eigenschaft definiert die Beziehung der ID (guid).

Rückgabewerte:

*BSTR** *pVal*

Beispiel:

```
BSTR* pVal = IDrtRelation.RelID
```

Siehe: [Allgemeine Beispiele](#)

HRESULT RelID ([in] BSTR *newVal*)

Beschreibung: Die Eigenschaft definiert die Beziehung der ID (guid).

Parameter:

BSTR *newVal*

Beispiel:

```
IDrtRelation.RelID(BSTR newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT RelTextID ([out, retval] BSTR * *pVal*)

Beschreibung: Die Eigenschaft definiert die Beziehung text id (guid) - Bezug auf die Zeile in der osRelText Tabelle.

Rückgabewerte:

*BSTR** *pVal*

Beispiel:

```
BSTR* pVal = IDrtRelation.RelTextID
```

Siehe: [Allgemeine Beispiele](#)

HRESULT RelTextID ([in] BSTR *newVal*)

Beschreibung: Die Eigenschaft definiert die Beziehung text id (guid) - Bezug auf die Zeile in der osRelText Tabelle.

Parameter:

BSTR *newVal*

Beispiel:

```
IDrtRelation.RelTextID(BSTR newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT IObjID ([out, retval] long * *pVal*)

Beschreibung: Die Eigenschaft definiert das object_id Feld vom Info-Objekt.

Rückgabewerte:

*LONG** pVal

Beispiel:

```
LONG* pVal = IDrtRelation.IObjID
```

Siehe: [Allgemeine Beispiele](#)

HRESULT IObjID ([in] long *newVal*)

Beschreibung: Die Eigenschaft definiert das object_id Feld vom Info-Objekt.

Parameter:

LONG newVal

Beispiel:

```
IDrtRelation.IObjID(LONG newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT IObjType ([out, retval] long * *pVal*)

Beschreibung: Die Eigenschaft definiert das object_type Feld vom Info-Objekt.

Rückgabewerte:

*LONG** pVal

Beispiel:

```
LONG* pVal = IDrtRelation.IObjType
```

Siehe: [Allgemeine Beispiele](#)

HRESULT IObjType ([in] long *newVal*)

Beschreibung: Die Eigenschaft definiert das object_type Feld vom Info-Objekt.

Parameter:

LONG newVal

Beispiel:

```
IDrtRelation.IObjType(LONG newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Created ([out, retval] long * *pVal*)

Beschreibung: Die Eigenschaft definiert den Zeitpunkt an dem die Beziehung angelegt wurde. Der Datentyp ist ein Zeitstempel: Anzahl der Sekunden seit Mitternacht (00:00:00), 1. Januar, 1970.

Rückgabewerte:

*LONG** pVal

Beispiel:

```
LONG* pVal = IDrtRelation.Created
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Created ([in] long *newVal*)

Beschreibung: Die Eigenschaft definiert den Zeitpunkt an dem die Beziehung angelegt wurde. Der Datentyp ist ein Zeitstempel: Anzahl der Sekunden seit Mitternacht (00:00:00), 1. Januar, 1970.

Parameter:

LONG *newVal*

Beispiel:

```
IDrtRelation.Created (LONG newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT CreatedBy ([out, retval] BSTR * *pVal*)

Beschreibung: Die Eigenschaft definiert den Namen des Benutzers, der die Beziehung angelegt hat.

Rückgabewerte:

*BSTR** *pVal*

Beispiel:

```
BSTR* pVal = IDrtRelation.CreatedBy
```

Siehe: [Allgemeine Beispiele](#)

HRESULT CreatedBy ([in] BSTR *newVal*)

Beschreibung: Die Eigenschaft definiert den Namen des Beziehungserstellers.

Parameter:

BSTR *newVal*

Beispiel:

```
IDrtRelation.CreatedBy (BSTR newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ModifyTime ([out, retval] long * *pVal*)

Beschreibung: Die Eigenschaft definiert die Zeitpunkt der letzten Änderung der Relation. Der Datentyp ist ein Zeitstempel: Anzahl der Sekunden seit Mitternacht (00:00:00), 1. Januar, 1970.

Rückgabewerte:

*LONG** *pVal*

Beispiel:

```
LONG* pVal = IDrtRelation.ModifyTime
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ModifyTime ([in] long *newVal*)

Beschreibung: Die Eigenschaft definiert den Zeitpunkt der letzten Änderung der Relation. Der Datentyp ist ein Zeitstempel: Anzahl der Sekunden seit Mitternacht (00:00:00), 1. Januar, 1970.

Parameter:

LONG *newVal*

Beispiel:

```
IDrtRelation.ModifyTime (LONG newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ModifyUser ([out, retval] BSTR * *pVal*)

Beschreibung: Die Eigenschaft definiert den Namen des Benutzers seit der letzten Modifikation.

Rückgabewerte:

*BSTR** *pVal*

Beispiel:

```
BSTR* pVal = IDrtRelation.ModifyUser
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ModifyUser ([in] BSTR *newVal*)

Beschreibung: Die Eigenschaft definiert den Namen des Benutzers seit der letzten Modifikation.

Rückgabewerte:

BSTR *newVal*

Beispiel:

```
IDrtRelation.ModifyUser (BSTR newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Deleted ([out, retval] long * *pVal*)

Beschreibung: Die Eigenschaften gibt den Zeitpunkt an, an dem die Relation gelöscht wurde. Der Datentyp ist ein Zeitstempel: Anzahl der Sekunden seit Mitternacht (00:00:00), 1. Januar, 1970.

Rückgabewerte:

*LONG** *pVal*

Beispiel:

```
LONG* pVal = IDrtRelation.Deleted
```

HRESULT DeleteUser ([out, retval] BSTR * *pVal*)

Beschreibung: Die Eigenschaft definiert den Namen des Benutzers, der die Relation gelöscht hat.

Rückgabewerte:

*BSTR** *pVal*

Beispiel:

```
BSTR* pVal = IDrtRelation.ModifyUser
```

IDrtRelationMgr Schnittstellenreferenz

Abgeleitet von IDispatch.

Ausführliche Beschreibung

Diese Schnittstelle beschreibt Methoden zur Bearbeitung von Relationen und Relationstext - Objekten. Es wird beschrieben, wie diese Objekte angelegt, bearbeitet und recherchiert werden können.

Öffentliche Methoden

HRESULT [AddRelText](#) ([in] long ObjType1,[in] long ObjType2,[in] BSTR ShortRel,[in] long Active,[in] BSTR TextTo,[in] BSTR TextReverse,[out] VARIANT *pRelTextID,[out, retval] long *pResult)

HRESULT [AddRelTextLang](#) ([in] BSTR RelTextID,[in] long ObjType1,[in] long ObjType2,[in] long LangID,[in] BSTR TextTo,[in] BSTR TextReverse,[out, retval] long *pResult)

HRESULT [ModRelText](#) ([in] BSTR RelTextID,[in] long ObjType1,[in] long ObjType2,[in] BSTR ShortRel,[in] long Active,[in] BSTR TextTo,[in] BSTR TextReverse,[out, retval] long *pResult)

HRESULT [ModRelTextLang](#) ([in] BSTR RelTextID,[in] long ObjType1,[in] long ObjType2,[in] long LangID,[in] BSTR TextTo,[in] BSTR TextReverse,[out, retval] long *pResult)

HRESULT [DelRelText](#) ([in] BSTR RelTextID,[in] long LangID,[out, retval] long *pResult)

HRESULT [RetrieveRelTexts](#) ([in] long ObjType1,[in] long ObjType2,[in] long Active,[in] long LangID,[out, retval] IDrtRelationTexts **ppRelationTexts)

HRESULT [AddRel](#) ([in] long Obj_id1,[in] long Obj_typ1,[in] long Obj_id2,[in] long Obj_typ2,[in] long Valid_from,[in] long Valid_to,[in] long Ensured,[in] long Checked,[in] BSTR RelTextID,[in] long IObjID,[in] long IObjType,[out] VARIANT *pRelID,[out, retval] long *pResult)

HRESULT [ModRel](#) ([in] BSTR RelID,[in] long Valid_from,[in] long Valid_to,[in] long Ensured,[in] long Checked,[in] BSTR RelTextID,[in] long IObjID,[in] long IObjType,[out, retval] long *pResult)

HRESULT [DelRel](#) ([in] BSTR RelID,[out, retval] long *pResult)

HRESULT [DelRelEx](#) ([in] BSTR RelID,[in] long lMode,[out, retval] long *pResult)

HRESULT [RetrieveRelations](#) ([in] long Obj_id1,[in] long Obj_typ1,[in] long Obj_id2,[in] long Obj_typ2,[in] long Direction,[in] long Valid_from,[in] long Valid_to,[in] long Ensured,[in] long Checked,[in] BSTR RelTextID,[in] long IObjID,[in] long IObjType,[in] long Created_from,[in] long Created_to,[in] long Modifytime_from,[in] long Modifytime_to,[out, retval] IDrtRelations **ppRelations)

HRESULT [RetrieveRelationsEx](#) ([in] long Obj_id1,[in] long Obj_typ1,[in] long Obj_id2,[in] long Obj_typ2,[in] long Direction,[in] long Valid_from,[in] long Valid_to,[in] long Ensured,[in] long Checked,[in] BSTR RelTextID,[in] long IObjID,[in] long IObjType,[in] long Created_from,[in] long Created_to,[in] long Modifytime_from,[in] long Modifytime_to,[in] long Deleted_from,[in] long Deleted_to,[out, retval] IDrtRelations **ppRelations)

Dokumentation der Elementfunktionen

HRESULT AddRelText ([in] long *ObjType1*, [in] long *ObjType2*, [in] BSTR *ShortRel*, [in] long *Active*, [in] BSTR *TextTo*, [in] BSTR *TextReverse*, [out] VARIANT * *pRelTextID*, [out, retval] long * *pResult*)

Beschreibung: Diese Methode erstellt ein neuen Relationstext in der Standardsprache. Diese Methode verwendet den Serverjob: dms.AddRelText.

Parameter:

LONG ObjType1: Objekttype vom from-object. Muss ein gültiger Objekttyp sein.

LONG ObjType2: Objekttype vom to-object. Muss ein gültiger Objekttyp sein.

BSTR ShortRel: Kurzbeschreibung. Darf nicht leer sein.

LONG Active: Eigenschaftsaktivität vom Relationstext. Muss 0 = false oder 1 = true sein.

BSTR TextTo: Relationstext von ObjType1 zum ObjType2. Darf nicht leer sein.

BSTR TextReverse: Relationstext vom ObjType2 zum ObjType1. Darf nicht leer sein.

*VARIANT** RelTextID: Ausgabeparamter - ID (guid) vom erstellten Relationstext. Paramter bekommt einen Rückgabewert zur Weiterverarbeitung.

Rückgabewerte:*LONG** pResult**Hinweis:**

pResult 0 - kein Fehler, oder dms.AddRelText job Fehlerrückgabe.

Beispiel:

```
LONG* pResult = DrtRelationMgr.AddRelText(LONG ObjType1, LONG ObjType2,
BSTR ShortRel, LONG Active, BSTR TextTo, BSTR TextReverse, VARIANT*RelTextID)
```

Siehe: [Allgemeine Beispiele](#)**Bemerkungen:**

Diese Eigenschaft kann gelesen und geschrieben werden.

HRESULT AddRelTextLang ([in] BSTR *RelTextID*, [in] long *ObjType1*, [in] long *ObjType2*, [in] long *LangID*, [in] BSTR *TextTo*, [in] BSTR *TextReverse*, [out, retval] long * *pResult*)

Beschreibung: Diese Methode fügt dem bestehenden Relationtext einen neuen Text hinzu in einer anderen Sprache als die Standardsprache. Diese Methode verwendet den Serverjob: dms.AddRelTextLang.**Parameter:***BSTR* RelTextID: id (guid) vom Relationtext in Standardsprache.*LONG* ObjType1: Objekttyp vom from-object. Muss ein gültiger Objekttyp sein.*LONG* ObjType2: Objekttyp vom to-object. Muss ein gültiger Objekttyp sein.*LONG* LangID: Sprach - ID. Muss eine Sprache sein die gültig im System ist und sich von der Standardsprache unterscheidet.*BSTR* TextTo: Relationtext vom ObjType1 zum ObjType2. Darf nicht leer sein.*BSTR* TextReverse: Relationtext vom ObjType2 zum ObjType1. Darf nicht leer sein.*VARIANT* pRelTextID: GUID des erzeugten Relationtext-Objektes.**Rückgabewerte:***LONG** pResult**Hinweis:**

pResult 0 - kein Fehler, oder dms.AddRelTextLang job Fehlerrückgabe.

Beispiel:

```
LONG* pResult = DrtRelationMgr.AddRelTextLang(BSTR RelTextID, LONG
ObjType1, LONG ObjType2, LONG LangID, BSTR TextTo, VARIANT TextReverse)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ModRelText ([in] BSTR *RelTextID*, [in] long *ObjType1*, [in] long *ObjType2*, [in] BSTR *ShortRel*, [in] long *Active*, [in] BSTR *TextTo*, [in] BSTR *TextReverse*, [out, retval] long * *pResult*)

Beschreibung: Diese Methode verändert den bestehenden Relationtext mit der Standardsprache. Diese Methode verwendet den Serverjob: dms.ModRelText.**Parameter:***BSTR* RelTextID: id (guid) vom zu modifizierten Relationtext in Standardsprache.*LONG* ObjType1: Objekttyp vom from-object. Muss ein gültiger Objekttyp sein.*LONG* ObjType2: Objekttyp vom to-object. Muss ein gültiger Objekttyp sein.*BSTR* ShortRel: Neue Kurzbeschreibung. Darf nicht leer sein. Ist das Feld nicht modifiziert dann muss dort der alte Wert rein.*LONG* Active: Eigenschaftsaktivität vom Relationtext. Muss 0 = false oder 1 = true sein.*BSTR* TextTo: Neuer Relationtext vom ObjType1 zum ObjType2. Darf nicht leer sein.

BSTR TextReverse: Neuer Relationstext vom ObjType2 zum ObjType1. Darf nicht leer sein.

Rückgabewerte:

*LONG** pResult:

Hinweis:

pResult 0 - kein Fehler, oder dms.ModRelText job Fehlerrückgabe.

Beispiel:

```
LONG* pResult = DrtRelationMgr.ModRelText(BSTR RelTextID, LONG ObjType1,
LONG ObjType2, BSTR ShortRel, LONG Active, BSTR TextTo, BSTR TextReverse)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ModRelTextLang ([in] BSTR *RelTextID*, [in] long *ObjType1*, [in] long *ObjType2*, [in] long *LangID*, [in] BSTR *TextTo*, [in] BSTR *TextReverse*, [out, retval] long * *pResult*)

Beschreibung: Diese Methode verändert den bestehenden Relationstext hinsichtlich der Standardsprache. Diese Methode verwendet den Serverjob: dms.ModRelTextLang.

Parameter:

BSTR RelTextID: id (guid) vom modifizierten Relationstext in Standardsprache.

LONG ObjType1: Objekttyp vom from-object. Muss ein gültiger Objekttyp sein.

LONG ObjType2: Objekttyp vom to-object. Muss ein gültiger Objekttyp sein.

LONG LangID: Sprach - ID. Muss gültig im System sein und sich unterscheiden vom Standard. Muss ein vorhandener relation text zum Übergeben von RelTextID und LangID.

BSTR TextTo: Neuer relation text vom ObjType1 zu ObjType2. Darf nicht leer sein.

BSTR TextReverse: Neuer relation text vom ObjType2 to ObjType1. Darf nicht leer sein.

Rückgabewerte:

*LONG** pResult

Hinweis:

pResult 0 - kein Fehler, oder dms.ModRelTextLang job Fehlerrückgabe.

Beispiel:

```
LONG* pResult = DrtRelationMgr.ModRelTextLang(BSTR RelTextID, LONG
ObjType1, LONG ObjType2, LONG LangID, BSTR TextTo, BSTR TextReverse)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT DelRelText ([in] BSTR *RelTextID*, [in] long *LangID*, [out, retval] long * *pResult*)

Beschreibung: Diese Methode löscht einen bestehenden Relationstext. Diese Methode verwendet den Serverjob: dms.DelRelText.

Parameter:

BSTR RelTextID: id (guid) vom zu löschenden relation text.

LONG LangID: Sprach - ID. Muss gültig im System sein oder 0 für Standard.

Es kann nur ein vorhandener Relationstext übergeben werden mit den entsprechenden Parametern RelTextID und LangID. Ist die LangID gleich der Standardsprache oder 0 - werden alle Einträge gelöscht mit der angegebenen RelTextID dessen Relationstexte mit dieser Sprach ID erstellt wurden.

Ist die LangID abweichend von dem Defaultwert dann löscht er nur diese Einträge.

Rückgabewerte:

*LONG** pResult:

Hinweis:

pResult 0 - keine Fehler, oder dms.DelRelText job Fehlerrückgabe.

Beispiel:

```
LONG* pResult = DrtRelationMgr.DelRelText(BSTR RelTextID, LONG LangID)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT RetrieveRelTexts ([in] long *ObjType1*, [in] long *ObjType2*, [in] long *Active*, [in] long *LangID*, [out, retval] [IDrtRelationTexts](#) ** *ppRelationTexts*)

Beschreibung: Diese Methode gibt eine Liste von Relationstexts zurück, die über die Schnittstelle [IDrtRelationTexts](#) ermittelt werden kann. Die Eingabeparameter werden benutzt um bei der Anfrageausführung spezifischer zu suchen.

Parameter:

LONG ObjType1: Objekttyp vom from-object.

LONG ObjType2: Objekttyp vom to-object.

LONG Active: Aktiver Status, 0 - false deaktiviert werden zurückgegeben, 1 - aktiviert werden zurückgegeben, -1 alle andere werden zurückgegeben

LONG LangID: Sprach ID. Muss im System gültig sein. 0 - ist gleich Standard. Es werden nur die Ergebnisse zurückgegeben die diese Lang ID besitzen.

Rückgabewerte:

*IDrtRelationTexts*** *ppRelationTexts*: Die Liste von abgerufenen RelationText

Hinweis: mögliche Varianten

Beispiel:

```
LONG* pResult = DrtRelationMgr.RetrieveRelTexts(LONG ObjType1, LONG
ObjType2, LONG Active, LONG LangID)
```

Siehe: [Allgemeine Beispiele](#)

RetrieveRelTexts(-1,-1,-1,0,ppRelationTexts) - abrufen aller Texte in der Standardsprache
 RetrievePortfolios(-1,-1,1,9,ppRelationTexts) - abrufen aller aktiven Texte mit der Sprach ID 9

HRESULT AddRel ([in] long *Obj_id1*, [in] long *Obj_typ1*, [in] long *Obj_id2*, [in] long *Obj_typ2*, [in] long *Valid_from*, [in] long *Valid_to*, [in] long *Ensured*, [in] long *Checked*, [in] BSTR *RelTextID*, [in] long *IObjID*, [in] long *IObjType*, [out] VARIANT* *pRelID*, [out, retval] long* *pResult*)

Beschreibung: Diese Methode erstellt eine neue Relation (Beziehung). Diese Methode verwendet den Serverjob: dms.AddRel.

Parameter:

LONG Obj_id1: Objekt ID vom from-object. Muss eine gültige Objekt ID sein.

LONG Obj_typ1: Objekttyp vom from-object. Muss ein gültiger Objekttyp sein.

LONG Obj_id2: Objekt ID vom to-object. Muss eine gültige Objekt ID sein.

LONG Obj_typ2: Objekttyp vom to-object. Muss ein gültiger Objekttyp sein.

LONG Valid_from: **Siehe auch:** [IDrtRelation](#) . Darf nicht leer sein.

LONG Valid_to: **Siehe auch:** [IDrtRelation](#) . Darf nicht leer sein.

LONG Ensured: **Siehe auch:** [IDrtRelation](#) . Muss 0 oder 1 sein.

LONG Checked: **Siehe auch:** [IDrtRelation](#) . Muss 0 oder 1 sein.

BSTR RelTextID: ID vom relation text vom Obj_typ1 zu Obj_typ2. Muss eine gültigen Bezug auf die Tabelle osRelText haben.

LONG IObjID: Objekt ID vom info-object. Muss eine gültige Objekt ID sein.

LONG IObjType: Objekttyp vom info-object. Muss ein gültiger Objekttyp sein.

*VARIANT** RelID: Ausgabeparameter ID (guid) von der erstellt Relation.

Rückgabewerte:

*LONG** pResult:

Hinweis: mögliche Fehlercodes

pResult 0 - keine Fehler, oder dms.AddRel job Fehlerrückgabe.

Beispiel:

```
LONG* pResult = DrtRelationMgr.AddRel(LONG Obj_id1, LONG Obj_typ1, LONG
Obj_id2, LONG Obj_typ2, LONG Valid_from, LONG Valid_to, LONG Ensured, LONG Checked, BSTR
RelTextID, LONG IObjID, LONG IObjType, VARIANT*RelID)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ModRel ([in] BSTR RelID, [in] long Valid_from, [in] long Valid_to, [in] long Ensured, [in] long Checked, [in] BSTR RelTextID, [in] long IObjID, [in] long IObjType, [out, retval] long * pResult)

Beschreibung: Diese Methode verändert bestehende Relationen. Diese Methode verwendet den Serverjob: dms.ModRel.

Parameter:

BSTR RelID: ID (guid) vom der zu verändernen Relation.

LONG Valid_from: Neuer Wert für Valid_from. Darf nicht leer sein.

LONG Valid_to: Neuer Wert für Valid_to. Darf nicht leer sein.

LONG Ensured: **Siehe auch:** [IDrtRelation](#). Muss 0 oder 1 sein.

LONG Checked: **Siehe auch:** [IDrtRelation](#). Muss 0 oder 1 sein.

BSTR RelTextID: ID vom Relationstext vom Obj_typ1 zu Obj_typ2. Muss eine gültigen Bezug auf die Tabelle osRelText haben.

LONG IObjID: Objekt ID vom Referenzobjekt. Muss eine gültige Objekt ID sein.

LONG IObjType: Objekttyp vom Referenzobjekt. Muss ein gültiger Objekttyp sein.

Rückgabewerte:

*LONG** pResult:

Hinweis: mögliche Fehlercodes

pResult 0 - keine Fehler, oder dms.ModRel job Fehlerrückgabe.

Beispiel:

```
LONG* pResult = DrtRelationMgr.ModRel(BSTR RelID, LONG Valid_from, LONG
Valid_to, LONG Ensured, LONG Checked, BSTR RelTextID, LONG IObjID, LONG IObjType)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT DelRel ([in] BSTR RelID, [out, retval] long * pResult)

Beschreibung: Diese Methode löscht eine bestehende Relation(Beziehung). Diese Relation wird mit dieser Funktion nicht physikalisch gelöscht. Diese Methode verwendet den Serverjob: dms.DelRel.

Parameter:

BSTR RelID: ID (guid) von der zu löschenden Relation.

Rückgabewerte:

*LONG** pResult:

Hinweis: mögliche Fehlercodes

pResult 0 - keine Fehler, oder dms.DelRel job Fehlerrückgabe.

Beispiel:

```
LONG* pResult = DrtRelationMgr.DelRel(RelID)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT DelRelEx ([in] BSTR RelID, [in] long lMode, [out, retval] long * pResult)

Beschreibung: Diese Methode stellt eine Erweiterung der Methode DelRel dar. Sie ermöglicht das Löschen einer bestehenden Relation(Beziehung). Je nach dem Modus wird die Relation physikalisch gelöscht oder nicht. Diese Methode verwendet den Serverjob: dms.DelRel.

Parameter:

BSTR RelID: ID (guid) von der zu löschenden Relation.

long lMode: Modus für das Löschen einer Relations (0: einfaches Löschen, 1: physikalisches Löschen, die Relation ist anschließend in der Datenbank gelöscht)

Rückgabewerte:

*LONG** pResult:

Hinweis: mögliche Fehlercodes

pResult 0 - keine Fehler, oder dms.DelRel job Fehlerrückgabe.

Beispiel:

```
LONG* pResult = DrtRelationMgr.DelRelEx(RelID,1)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT RetrieveRelations ([in] long Obj_id1, [in] long Obj_typ1, [in] long Obj_id2, [in] long Obj_typ2, [in] long Direction, [in] long Valid_from, [in] long Valid_to, [in] long Ensured, [in] long Checked, [in] BSTR RelTextID, [in] long IObjID, [in] long IObjType, [in] long Created_from, [in] long Created_to, [in] long Modifytime_from, [in] long Modifytime_to, [out, retval] IDrtRelations ** ppRelations)

Beschreibung: Diese Methode ermöglicht das Recherchieren nach Relationen, die im System angelegt sind. Über die Eingangsparameter dieser Methode kann die Recherche parametrisiert werden, um mit Hilfe bestimmter Kriterien nach Relationen im System zu suchen. Diese Methode verwendet den Serverjob: dms.RetrieveRelations.

Parameter:

LONG Obj_id1: Objekt ID vom from-object. Bei -1 werden alle from-object Ids gesucht.

LONG Obj_typ1: Objekttyp vom from-object. Bei -1 werden alle from-object Typen gesucht.

LONG Obj_id2: Objekt ID vom to-object. Bei -1 werden alle to-object Ids gesucht.

LONG Obj_typ2: Objekttyp vom to-object. Bei -1 werden alle to-object Typen gesucht.

LONG Direction: 0 oder 1. Ist 1 - abrufen in beide Richtungen: von 1 zu 2 und von 2 zu 1.

LONG Valid_from, Valid_to - werden nicht benutzt

LONG Ensured: ist Wert gleich 0 - abrufen nicht sichergestellter Relationen, 1 abrufen aller sichergestellten Relationen, -1 abrufen aller Relationen

LONG Checked: ist Wert gleich 0 - Abruf nicht gecheckter R., 1 - Abruf gecheckter R., -1 Abruf aller R.

BSTR RelTextID, *LONG* IObjID, *LONG* IObjType - werden nicht benutzt

LONG Created_from: Untergrenze von der Erstellzeit. Ist 0 - wird diese Option nicht benutzt.

LONG Created_to: Obergrenze von der Erstellzeit. Ist 0 - wird diese Option nicht benutzt.

LONG Modifytime_from: Untergrenze der Modifikationszeit. Ist 0 - wird diese Option nicht benutzt.

LONG Modifytime_to: Obergrenze der Modifikationsgrenze. Ist 0 - wird diese Option nicht benutzt.

Rückgabewerte:

*IDrtRelations*** ppRelations: Liste der abgerufenen Relationen.

RetrieveRelations(-1,-1,-1,-1,1,0,0,-1,-1,"",-1,-1,0,0,0,0,ppRelations) - Abruf aller Relationen

RetrieveRelations(-1,-1,-1,-1,1,0,0, 1, 0,"",-1,-1,0,0,0,0,ppRelations) - Abruf aller sichergestellten, nicht gecheckten Relationen

RetrieveRelations(-1,11111,-1,-1,1,0,0,-1,-1,"",-1,-1,0,0,0,0,ppRelations) - Abruf der Relationen von allen Objekten mit dem Typ 11111 zu allen Objekten mit dem selben Typ und von allen Objekten mit dem selben Typ zu allen Objekten mit dem Typ 11111. (Richtung = 1).

Beispiel:

```
IDrtRelations** ppRelations = DrtRelationMgr.RetrievesRelations(LONG
Obj_id1, LONG Obj_typ1, Obj_id2, Obj_typ2, LONG Valid_from, LONG Valid_to, LONG Ensured, LONG
```

```
Checked, BSTR RelTextID, LONG IObjID, LONG IObjType, LONG Created_from, LONG Created_to, LONG
Modifytime_from, LONG Modifytime_to, LONG Deleted)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT RetrieveRelationsEx ([in] long *Obj_id1*, [in] long *Obj_typ1*, [in] long *Obj_id2*, [in] long *Obj_typ2*, [in] long *Direction*, [in] long *Valid_from*, [in] long *Valid_to*, [in] long *Ensured*, [in] long *Checked*, [in] BSTR *RelTextID*, [in] long *IObjID*, [in] long *IObjType*, [in] long *Created_from*, [in] long *Created_to*, [in] long *Modifytime_from*, [in] long *Modifytime_to*, [in] long *Deleted_from*, [in] long *Deleted_to*, [out, retval] [IDrtRelations](#) ** *ppRelations*)

Beschreibung: Diese Methode erweitert die Methode RetrieveRelations. Sie ermöglicht das Recherchieren nach gelöschten oder (physikalisch) nicht gelöschten Relationen, die im System angelegt sind. Über die Eingangsparameter dieser Methode kann die Recherche parametrisiert werden, um mit Hilfe bestimmter Kriterien nach Relationen im System zu suchen. Diese Methode verwendet den Serverjob: dms.RetrieveRelations.

Parameter:

*LONG*Obj_id1: Objekt ID vom from-object. Bei -1 werden alle from-object Ids gesucht.
*LONG*Obj_typ1: Objekttyp vom from-object. Bei -1 werden alle from-object Typen gesucht.
*LONG*Obj_id2: Objekt ID vom to-object. Bei -1 werden alle to-object Ids gesucht.
*LONG*Obj_typ2: Objekttyp vom to-object. Bei -1 werden alle to-object Typen gesucht.
*LONG*Direction: 0 oder 1. Ist 1 - abrufen in beide Richtungen: von 1 zu 2 und von 2 zu 1.
*LONG*Valid_from, Valid_to - werden nicht benutzt
*LONG*Ensured: ist Wert gleich 0 - abrufen nicht sichergestellter Relationen, 1 abrufen aller sichergestellten Relationen, -1 abrufen aller Relationen
*LONG*Checked: ist Wert gleich 0 - Abruf nicht gecheckter R., 1 - Abruf gecheckt R., -1 Abruf aller R.
BSTR RelTextID, *LONG* IObjID, *LONG* IObjType - werden nicht benutzt
*LONG*Created_from: Untergrenze von der Erstellzeit. Ist 0 - wird diese Option nicht benutzt.
*LONG*Created_to: Obergrenze von der Erstellzeit. Ist 0 - wird diese Option nicht benutzt.
*LONG*Modifytime_from: Untergrenze der Modifikationszeit. Ist 0 - wird diese Option nicht benutzt.
*LONG*Modifytime_to: Obergrenze der Modifikationszeit. Ist 0 - wird diese Option nicht benutzt.
*LONG*Deleted_from: Untere Zeitgrenze für die Suche nach gelöschten Relationen. Ist 0 - wird diese Option nicht benutzt, es werden dann alle nicht gelöschten Relationen zurückgegeben
*LONG*Deleted_to: Obere Zeitgrenze für die Suche nach gelöschten Relationen. Ist 0 - wird Modifytime_from als exakte Zeit für die Suche genommen. Bei -1 wird Deleted_from als untere Grenze der Suche genommen. Die obere Grenze ist offen. Der Spezialfall Deleted_from = 0 und Deleted_to = -1 liefert alle gelöschten und nicht gelöschten Relation im System.

Rückgabewerte:

*IDrtRelations*** *ppRelations*: Liste der abgerufenen Relationen.

RetrieveRelationsEx(-1,-1,-1,-1,1,0,0,-1,-1,"",-1,-1,0,0,0,0,1,-1,ppRelations) - Abruf aller gelöschten Relationen

RetrieveRelationsEx(-1,-1,-1,-1,1,0,0, 1, 0,"",-1,-1,0,0,0,0,0,0,ppRelations) - Abruf aller nicht gelöschten, sichergestellten, nicht gecheckten Relationen

RetrieveRelationsEx(-1,11111,-1,-1,1,0,0,-1,-1,"",-1,-1,0,0,0,0,0,0,ppRelations) - Abruf der Relationen von allen Objekten mit dem Typ 11111 zu allen Objekten mit dem selben Typ und von allen Objekten mit dem selben Typ zu allen Objekten mit dem Typ 11111. (Richtung = 1). Es werden nur ungelöschte Relationen zurückgegeben.

Beispiel:

```
IDrtRelations** ppRelations = DrtRelationMgr.RetrievesRelationsEx(LONG
Obj_id1, LONG Obj_typ1, Obj_id2, Obj_typ2, LONG Valid_from, LONG Valid_to, LONG Ensured, LONG
Checked, BSTR RelTextID, LONG IObjID, LONG IObjType, LONG Created_from, LONG Created_to, LONG
Modifytime_from, LONG Modifytime_to, LONG Deleted)
```

Siehe: [Allgemeine Beispiele](#)

IDrtRelations Schnittstellenreferenz

Abgeleitet von IDispatch.

Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Zusammenstellung von [IDrtRelation](#) Objekte.

Siehe auch: [IDrtRelation](#)

Öffentliche Methoden

HRESULT [Deserialize](#) ([in] BSTR bstrXml)

HRESULT [Serialize](#) ([out, retval] BSTR *pXml)

Dokumentation der Elementfunktionen

HRESULT Deserialize ([in] BSTR *bstrXml*)

Beschreibung: Diese Methode setzt Objekteigenschaften durch einen XML formatierten String.

Parameter:

BSTR bstrXml: xml-input string

Siehe auch: [Relations Objekte](#)

Beispiel:

```
IDrtRelations.Deserialize (BSTR bstrXml)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Serialize ([out, retval] BSTR **pXml*)

Beschreibung: Diese Methode erhält eine Objekteigenschaftsbeschreibung als XML formatierten String.

Rückgabewerte:

*BSTR** pXml: output xml-string

Beispiel:

```
BSTR* pXml = IDrtRelations.Serialize
```

Siehe: [Allgemeine Beispiele](#)

IDrtRelationText Schnittstellenreferenz

Abgeleitet von IDispatch.

Ausführliche Beschreibung

Diese Schnittstelle stellt mit Hilfe einer Datenbanktabelle die RelationText Objekte dar.

Öffentliche Methoden

HRESULT [Deserialize](#) ([in] BSTR bstrXml)
 HRESULT [Serialize](#) ([out, retval] BSTR *pXml)
 HRESULT [RelTextID](#) ([out, retval] BSTR *pVal)
 HRESULT [RelTextID](#) ([in] BSTR newVal)
 HRESULT [ShortRel](#) ([out, retval] BSTR *pVal)
 HRESULT [ShortRel](#) ([in] BSTR newVal)
 HRESULT [TextTo](#) ([out, retval] BSTR *pVal)
 HRESULT [TextTo](#) ([in] BSTR newVal)
 HRESULT [TextReverse](#) ([out, retval] BSTR *pVal)
 HRESULT [TextReverse](#) ([in] BSTR newVal)
 HRESULT [Active](#) ([out, retval] long *pVal)
 HRESULT [Active](#) ([in] long newVal)
 HRESULT [ObjType1](#) ([out, retval] long *pVal)
 HRESULT [ObjType1](#) ([in] long newVal)
 HRESULT [ObjType2](#) ([out, retval] long *pVal)
 HRESULT [ObjType2](#) ([in] long newVal)

Dokumentation der Elementfunktionen

HRESULT Deserialize ([in] BSTR *bstrXml*)

Beschreibung: Diese Methode füllt die RelationText Felder durch den XML- formatierten String.

Parameter:

BSTR bstrXml: xml-input string

Beispiel:

```
IDrtRelationText.Deserialize (BSTR bstrXml)
```

Siehe: [Allgemeine Beispiele](#) , RelationText

HRESULT Serialize ([out, retval] BSTR **pXml*)

Beschreibung: Diese Methode erhält eine XML Beschreibung von den Objekteigenschaften.

Rückgabewerte:

*BSTR** pXml: output xml-string

Beispiel:

```
BSTR* pXml = IDrtRelationText.Serialize
```

Siehe: [Allgemeine Beispiele](#)

HRESULT RelTextID ([out, retval] BSTR * *pVal*)

Beschreibung: Diese Methode erhält eine Textbeschreibung des Relationtextes.

Rückgabewerte:

*BSTR** *pVal*

Beispiel:

```
BSTR* pVal = IDrtRelationText.RelTextID
```

Siehe: [Allgemeine Beispiele](#)

HRESULT RelTextID ([in] BSTR *newVal*)

Beschreibung: Diese Methode definiert eine neue Textbeschreibung für diesen Relationtext.

Parameter:

BSTR *newVal*

Beispiel:

```
IDrtRelationText.RelTextID(BSTR newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ShortRel ([out, retval] BSTR * *pVal*)

Beschreibung: Diese Methode erhält eine kurze eindeutige Textbeschreibung für Relationstexte.

Rückgabewerte:

*BSTR** *pVal*

Beispiel:

```
BSTR* pVal = IDrtRelationText.ShortRel
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ShortRel ([in] BSTR *newVal*)

Beschreibung: Diese Methode definiert eine neue kurze eindeutige Textbeschreibung für Relationstexte.

Parameter:

BSTR *newVal*

Beispiel:

```
IDrtRelationText.ShortRel(BSTR newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT TextTo ([out, retval] BSTR * *pVal*)

Beschreibung: Diese Methode erhält eine Textbeschreibung der Relation von Objekttyp1 zu Objekttyp2.

Rückgabewerte:

*BSTR** *pVal*

Beispiel:

```
BSTR* pVal = IDrtRelationText.TextTo
```

Siehe: [Allgemeine Beispiele](#)

HRESULT TextTo ([in] BSTR *newVal*)

Beschreibung: Diese Methode erstellt eine Textbeschreibung der Relation von Objekttyp1 zu Objekttyp2.

Parameter:

BSTR newVal

Beispiel:

```
IDrtRelationText.TextTo (BSTR newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT TextReverse ([out, retval] BSTR * *pVal*)

Beschreibung: Diese Methode erhält eine Textbeschreibung der Relation von Objekttyp2 zu Objekttyp1.

Rückgabewerte:

*BSTR** pVal

Beispiel:

```
BSTR* pVal = IDrtRelationText.TextReverse
```

Siehe: [Allgemeine Beispiele](#)

HRESULT TextReverse ([in] BSTR *newVal*)

Beschreibung: Diese Methode erhält eine Textbeschreibung der Relation von Objekttyp2 zu Objekttyp1.

Parameter:

BSTR newVal

Beispiel:

```
IDrtRelationText.TextReverse (BSTR newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Active ([out, retval] long * *pVal*)

Beschreibung: Die Eigenschaft definiert den Aktivstatus vom Beziehungstext. 0 - nicht aktiv, 1 - aktiv

Rückgabewerte:

*LONG** pVal

Beispiel:

```
LONG* pVal = IDrtRelationText.Active
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Active ([in] long *newVal*)

Beschreibung: Die Eigenschaft definiert den Aktivstatus vom Beziehungstext. 0 - nicht aktiv, 1 - aktiv

Parameter:

LONG *newVal*

Beispiel:

```
IDrtRelationText.Active (LONG newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ObjType1 ([out, retval] long * *pVal*)

Beschreibung: Diese Methode erhält den Typ vom ersten Objekt was zum entsprechenden Relationtext gehört.

Rückgabewerte:

*LONG** *pVal*

Beispiel:

```
LONG* pVal = IDrtRelationText.ObjType1
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ObjType1 ([in] long *newVal*)

Beschreibung: Diese Methode erstellt den Typ vom ersten "from" Objekt was zum entsprechenden Relationtext gehört.

Parameter:

LONG *newVal*

Beispiel:

```
IDrtRelationText.ObjType1 (LONG newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ObjType2 ([out, retval] long * *pVal*)

Beschreibung: Diese Methode erhält den Typ vom zweiten "to" Objekt was zum entsprechenden Relationstext gehört.

Rückgabewerte:

*LONG** *pVal*

Beispiel:

```
LONG* pVal = IDrtRelationText.ObjType2
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ObjType2 ([in] long *newVal*)

Beschreibung: Diese Methode erstellt den Typ vom zweiten "to" Objekt was zum entsprechenden Relationstext gehört.

Parameter:*LONG*newVal**Beispiel:**

```
IDrtRelationText.ObjType2 (LONG newVal)
```

Siehe: [Allgemeine Beispiele](#)

IDrtRelationTexts Schnittstellenreferenz

Abgeleitet von IDispatch.

Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Zusammenstellung von [IDrtRelationText](#) Objekte.

Siehe auch: [IDrtRelationText](#)

Öffentliche Methoden

HRESULT [Deserialize](#) ([in] BSTR bstrXml)

HRESULT [Serialize](#) ([out, retval] BSTR *pXml)

Dokumentation der Elementfunktionen

HRESULT Deserialize ([in] BSTR *bstrXml*)

Beschreibung: Diese Methode setzt Objekteigenschaften anhand einer XML Beschreibung.

Parameter:

BSTR bstrXml: xml-input string

Beispiel:

```
IDrtRelationTexts.Deserialize(BSTR bstrXml)
```

Siehe: [Allgemeine Beispiele](#) , RelationTexts

HRESULT Serialize ([out, retval] BSTR **pXml*)

Beschreibung: Diese Methode erhält eine XML Beschreibung von den Objekteigenschaften

Rückgabewerte:

*BSTR** pXml: output xml-string

Beispiel:

```
BSTR* pXml = IDrtRelationTexts.Serialize
```

Siehe: [Allgemeine Beispiele](#)

IDrtResult Schnittstellenreferenz

Abgeleitet von IDispatch.

Ausführliche Beschreibung

Diese Schnittstelle beschreibt ein einzelnes Objekt der Ergebnisliste einer Datenanfrage.

Öffentliche Methoden

HRESULT [ObjectType](#) ([out, retval] long *plType)
HRESULT [ID](#) ([out, retval] long *plID)
HRESULT [Name](#) ([out, retval] BSTR *pbstrName)
HRESULT [GetFieldCount](#) ([out, retval] long *plCount)
HRESULT [GetFieldName](#) ([in] long lIndex,[out, retval] BSTR *pbstrFieldName)
HRESULT [GetFieldValue](#) ([in] long lIndex,[out, retval] BSTR *pbstrFieldValue)

Dokumentation der Elementfunktionen

HRESULT ObjectType ([out, retval] long * *plType*)

Beschreibung: Diese Methode gibt den Objekttyp eines Ergebnisobjektes zurück.

Rückgabewerte:

*LONG** plType

Beispiel:

```
LONG* plType = DrtResult.ObjectType
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ID ([out, retval] long * *plID*)

Beschreibung: Diese Methode gibt die Objekt-ID des Ergebnisobjektes zurück.

Rückgabewerte:

*LONG** plID

Beispiel:

```
LONG* plID = DrtResult.ID
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Name ([out, retval] BSTR * *pbstrName*)

Beschreibung: Diese Methode gibt den Namen des Ergebnisobjektes zurück.

Rückgabewerte:

*BSTR** pbstrName: Name des Ergebnisobjektes

Beispiel:

```
BSTR* pbstrName = DrtResult.Name
```

Siehe: [Allgemeine Beispiele](#)

HRESULT GetFieldCount ([out, retval] long * *plCount*)

Beschreibung: Diese Methode gibt die Anzahl der Felder bzw. Feldwerte des Ergebnisobjektes zurück.

Rückgabewerte:

*LONG**plCount

Beispiel:

```
LONG* plCount = DrtResult.GetFieldCount
```

Siehe: [Allgemeine Beispiele](#)

HRESULT GetFieldName ([in] long *lIndex*, [out, retval] BSTR * *pbstrFieldName*)

Beschreibung: Diese Methode gibt vom Ergebnisobjekt den Namen des Feldes zurück.

Parameter:

*LONG*lIndex: Index des angefragten Feldes.

Rückgabewerte:

*BSTR**pbstrFieldName: Name des Feldes.

Beispiel:

```
BSTR*lbstrFieldName = DrtResult.GetFieldName(LONG lIndex)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT GetFieldValue ([in] long *lIndex*, [out, retval] BSTR * *pbstrFieldValue*)

Beschreibung: Diese Methode gibt einen Feldwert vom Ergebnisobjekts zurück.

Parameter:

*LONG*lIndex: Index des angefragten Feldes.

Rückgabewerte:

*BSTR**pbstrFieldValue: Wert des Feldes.

Beispiel:

```
BSTR* pbstrFieldValue = DrtResult.GetFieldValue(LONG lIndex)
```

Siehe: [Allgemeine Beispiele](#)

IDrtResults Schnittstellenreferenz

Abgeleitet von IDispatch.

Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Zusammenstellung von Objekten vom Typ [IDrtResult](#).

Siehe auch: [IDrtResult](#)

IDrtSecurity Schnittstellenreferenz

Abgeleitet von IDispatch.

Ausführliche Beschreibung

Diese Schnittstelle ermöglicht den Zugriff auf die Benutzer und Benutzergruppen des Systems. Es werden Serverjobs des Namespaces Admin aufgerufen.

Öffentliche Methoden

HRESULT [GetGroupList](#) ([out, retval] [IDrtGroups](#) **ppGroups)
 HRESULT [GetGroupAttributes](#) ([in] BSTR GroupName,[in, out] [IDrtGroup](#) *pGroup)
 HRESULT [GetUsersOfGroup](#) ([in] [IDrtGroup](#) *pGroup,[out, retval] [IDrtUsers](#) **ppUsers)
 HRESULT [GetUserList](#) ([out, retval] [IDrtUsers](#) **ppUsers)
 HRESULT [GetUserAttributes](#) ([in] BSTR UserName,[in, out] [IDrtUser](#) *pUser)
 HRESULT [GetGroupsOfUser](#) ([in] [IDrtUser](#) *pUser,[out, retval] [IDrtGroups](#) **ppGroups)

Dokumentation der Elementfunktionen

HRESULT GetGroupList ([out, retval] [IDrtGroups](#) ** ppGroups)

Beschreibung: Diese Methode liefert eine Liste aller Benutzergruppen des Systems zurück.

Rückgabewerte:

*IDrtGroups**ppGroups*: Liste der Benutzergruppen

Beispiel:

```
ppGroups = IDrtSecurity.GetGroupList
```

Siehe: [Allgemeine Beispiele](#)

HRESULT GetGroupAttributes ([in] BSTR GroupName,[in, out] [IDrtGroup](#) * pGroup)

Beschreibung: Diese Methode liefert die Eigenschaften einer Benutzergruppe in einem [IDrtGroup](#) - Objekt zurück.

Parameter:

BSTR GroupName: Gruppenname

*IDrtGroup*pGroup*: Objekt zu bereits existierenden Zuständen von [IDrtGroup](#)

Beispiel:

```
Dim pGroup
Set pGroup = CreateObject("axcdl.DrtGroup")
IDrtSecurity.GetGroupAttributes(BSTR GroupName, pGroup)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT GetUsersOfGroup ([in] [IDrtGroup](#) * pGroup, [out, retval] [IDrtUsers](#) ** ppUsers)

Beschreibung: Diese Methode liefert eine Liste mit den Benutzer einer Benutzergruppe (Es werden bei dieser Anfrage aus Performancegründen aber nicht alle Eigenschaften der Benutzer initialisiert. Auf folgende Benutzereigenschaften kann anschliessend zugegriffen werden: Name, Benutzer, ID, GUID).

Parameter:

*IDrtGroup**pGroup: Objekt der Gruppen, von welcher man die Mitglieder sucht.

Rückgabewerte:

*IDrtUsers***ppGroups: Liste der gefundenen Benutzer

Beispiel:

```
Dim pGroup  
  
Set pGroup = CreateObject("axcdl.DrtGroup")  
  
ppGroups = IDrtSecurity.GetUsersOfGroup(pGroup)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT GetUserList ([out, retval] [IDrtUsers](#) ** ppUsers)

Beschreibung: Die Methode liefert eine Liste mit allen Benutzern des Systems zurück. (Es werden bei dieser Anfrage aus Performancegründen aber nicht alle Eigenschaften der Benutzer initialisiert. Auf folgende Benutzereigenschaften kann anschliessend zugegriffen werden: Name, Benutzer, ID, GUID).

Rückgabewerte:

*IDrtUsers***ppUsers: Liste von Benutzern

Beispiel:

```
ppUsers = IDrtSecurity.GetUserList
```

Siehe: [Allgemeine Beispiele](#)

HRESULT GetUserAttributes ([in] BSTR UserName, [in, out] [IDrtUser](#) * pUser)

Beschreibung: Die Methode gibt die Eigenschaften eines Benutzers in einem [IDrtUser](#) - Objekt zurück.

Parameter:

BSTR UserName: Benutzername (Wert vom "benutzer" Feld in der Benutzer DB Tabelle),

*IDrtUser**pUser: Zeiger zu bereits existierenden Zuständen von [IDrtUser](#), welche Felder zu befüllen sind.

Beispiel:

```
Dim pUser  
  
Set pUser = CreateObject("axcdl.DrtUser")  
  
IDrtSecurity.GetUserAttributes(BSTR UserName, IDrtUser*pUser)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT GetGroupsOfUser ([in] [IDrtUser](#) * pUser, [out, retval] [IDrtGroups](#) ** ppGroups)

Beschreibung: Die Methode gibt eine Liste von Benutzergruppen zurück, denen der angegebene Benutzer angehört.

Parameter:

*IDrtUser**pUser: Objekte von Benutzern, von welchen man die Gruppen sucht.

Rückgabewerte:

*IDrtGroups***ppGroups: Liste von Gruppen

Beispiel:

```
Dim pUser  
Set pUser = CreateObject("axcdl.DrtUser")  
ppGroups = IDrtSecurity.GetGroupsOfUser(IDrtUser*pUser)
```

Siehe: [Allgemeine Beispiele](#)

IDrtSession Schnittstellenreferenz

Abgeleitet von IDispatch.

Ausführliche Beschreibung

Diese Schnittstelle verwaltet die Verbindung zum Server. Durch das Vorhandensein der entsprechenden Variablen werden diese von der [IDrtSession](#) initialisiert.

Öffentliche Methoden

HRESULT [Login](#) ([in] BSTR bstrUserName,[in] BSTR bstrPassword,[in] BSTR bstrServerName,[in] BSTR bstrServerPort,[in] BSTR bstrAppName,[in] BSTR bstrLicense,[in] LONG hWnd,[in, defaultvalue(1)] LONG IDbCursorCount)

HRESULT [ReuseLogin](#) (LONG IUnused)

HRESULT [Logout](#) ()

HRESULT [Connected](#) ([out, retval] VARIANT_BOOL *pbConnected)

HRESULT [UserName](#) ([out, retval] BSTR *pbstrUserName)

HRESULT [ServerName](#) ([out, retval] BSTR *pbstrServerName)

HRESULT [ServerPort](#) ([out, retval] BSTR *pbstrServerPort)

HRESULT [AppName](#) ([out, retval] BSTR *pbstrAppName)

HRESULT [License](#) ([out, retval] BSTR *pbstrLicense)

HRESULT [hWnd](#) ([out, retval] long *phWnd)

HRESULT [DbCursorCount](#) ([out, retval] long *plCount)

HRESULT [GetObjDefXml](#) ([out, retval] BSTR *pbstrXml)

HRESULT [GetResultFromXml](#) ([in] BSTR bstrXml,[out, retval] [IDrtResults](#) **ppResults)

HRESULT [GetResult](#) ([in] [IDrtQueries](#) *pQueries,[out, retval] [IDrtResults](#) **ppResults)

HRESULT [GetResultObjectsFromXml](#) ([in] BSTR bstrXml,[out, retval] [IDrtObjects](#) **ppResultObjects)

HRESULT [GetResultObjects](#) ([in] [IDrtQueries](#) *pQueries,[out, retval] [IDrtObjects](#) **ppResultObjects)

HRESULT [InsertFromXml](#) ([in] BSTR bstrXml,[in] DrtImportFlagEnum Flags,[out, retval] long *plObjectID)

HRESULT [Insert](#) ([in, out] [IDrtObject](#) *pObject,[in] DrtImportFlagEnum Flags,[out, retval] long *plObjectID)

HRESULT [UpdateFromXml](#) ([in] BSTR bstrXml,[in] DrtImportFlagEnum Flags)

HRESULT [Update](#) ([in] [IDrtObject](#) *pObject,[in] DrtImportFlagEnum Flags)

HRESULT [DeleteFromXml](#) ([in] BSTR bstrXml,[in] DrtImportFlagEnum Flags)

HRESULT [Delete](#) ([in] [IDrtObject](#) *pObject,[in] DrtImportFlagEnum Flags)

HRESULT [MoveFromXml](#) ([in] BSTR bstrXml,[in] DrtImportFlagEnum Flags)

HRESULT [Move](#) ([in] [IDrtObject](#) *pObject,[in] DrtImportFlagEnum Flags)

HRESULT [CreateRelationMgr](#) ([out, retval] [IDrtRelationMgr](#) **ppRelationMgr)

HRESULT [CreatePortfolioMgr](#) ([out, retval] [IDrtPortfolioMgr](#) **ppPortfolioMgr)

HRESULT [CreateAccessObj](#) ([out, retval] [IDrtAccess](#) **ppAccess)

HRESULT [CreateSecurityObj](#) ([out, retval] [IDrtSecurity](#) **ppSecurity)

HRESULT [CreateUserMgr](#) ([out, retval] [IDrtUserMgr](#) **ppUserMgr)

HRESULT [CreateJob](#) ([in]BSTR bsNamespace,[in]BSTR bsFuncname,[out, retval] IDrtJob **ppJob)

HRESULT [ResetWorkTime](#) ()

HRESULT [ResetJobTime](#) ()

HRESULT [WorkTime](#) ([out, retval] long *plTime)

HRESULT [JobTime](#) ([out, retval] long *plTime)

HRESULT [LoadObjectFiles](#) ([in] [IDrtObject](#) *pObject,[in] DrtLoadObjFileFlagsEnum enFlags,[out, retval] BSTR *pbstrFiles)

HRESULT [SaveObjectFiles](#) ([in] [IDrtObject](#) *pObject,[in] DrtSaveObjFileFlagsEnum enFlags,[in] BSTR bstrFiles,[out, retval] long *plSize)

HRESULT [LastError](#) ([out, retval] BSTR *pVal)

HRESULT [ErrorDetails](#) ([out, retval] BSTR *pVal)

HRESULT [LastHits](#) ([out, retval] long *pVal)

HRESULT [UserID](#) ([out, retval] long *pVal)

HRESULT [OSLocalWorkDir](#) ([out, retval] BSTR *bstrPath)

HRESULT [GetCurrentUserData](#) ([in] BSTR bstrName,[in] long lType,[out, retval] BSTR *bstrData)

HRESULT [ShallShowLoginDlg](#) ([out, retval] VARIANT_BOOL *pVal)

HRESULT [ShallShowLoginDlg](#) ([in] VARIANT_BOOL newVal)

HRESULT [CheckLicense](#) ([in] BSTR bsLicense,[out, retval] long *lRet)

HRESULT [GetDeletedObjects](#) ([in, optional, defaultvalue(0)] long IUserId,[in, optional, defaultvalue(NULL)] [IDrtQueries](#) *pQueries,[out, retval] [IDrtObjects](#) **ppResultObjects)
 HRESULT [RestoreObject](#) ([in] [IDrtObject](#) *pObject,[in, optional, defaultvalue(DrtRestore_Single)] DrtRestoreFlagsEnum flags)

Dokumentation der Elementfunktionen

HRESULT Login ([in] BSTR *bstrUserName*, [in] BSTR *bstrPassword*, [in] BSTR *bstrServerName*, [in] BSTR *bstrServerPort*, [in] BSTR *bstrAppName*, [in] BSTR *bstrLicense*, [in] LONG *hWnd*, [in, defaultvalue(1)] LONG *lDbCursorCount*)

Beschreibung: Die Methode ermöglicht das Anmelden eines Benutzers am enaio® Applikations - Server.

Hinweis:

Erst wenn ein Login vollzogen wurde können die weiteren Methoden verwendet werden. Dort braucht man das DrtSession Object nicht neu erstellen.

Dazu müssen folgende Parameter angegeben werden:

Parameter:

BSTR *bstrUserName* Name des anzumeldenden Benutzers
BSTR *bstrPassword* Passwort des anzumeldenden Benutzers
BSTR *bstrServerName* Rechnername des optimal_AS® Servers
BSTR *bstrServerPort* Port des Servers
BSTR *bstrAppName* Name der Applikation, die die Anmeldung durchführt
BSTR *bstrLicense* Lizenzstring der Applikation, die die Anmeldung
LONG *hWnd* Fensterhandle der Applikation
LONG *lDbCursorCount* Anzahl der zur Verfügung stehenden Datenbank-Cursor(default=1)

Bemerkungen:

Bei der Anmeldung werden die entsprechenden Variablen von [IDrtSession](#) initialisiert.

Beispiel:

```
Dim DrtSession //Object was erstellt werden muss

Set DrtSession = CreateObject("axcdl.DrtSession")

DrtSession.Login BSTR bstrUserName, BSTR bstrPassword, BSTR bstrServerName, BSTR
bstrServerPort, BSTR bstrAppName, BSTR bstrLicence, LONG hWnd, LONG lDbCursorCount
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ReuseLogin (LONG *lUnused*)

Beschreibung: Die Methode ermöglicht das Anmelden der Komponente am enaio® Applikations - Server bei bestehender Serververbindung durch das aufrufen einer Client Applikation.

Parameter:

LONG *lUnused* Zeiger auf die Context - Struktur der Serververbindung

Rückgabewerte:

lResult

Hinweis: mögliche Fehlercodes

- 0 - kein Fehler
- 16 - Fehler beim Verbindungsaufbau
- 256 - Das Datenbankmodul konnte nicht geladen werden
- 273 - Abbruch durch Benutzer (beim Serverauswahldialog)
- 469 - Benutzeranmeldung fehlgeschlagen
- 470 - Lizenzierung fehlgeschlagen
- 4096 - Fehler beim Lesen von Lizenzinformationen

Bemerkungen:

Bei der Anmeldung werden die entsprechenden Variablen von [IDrtSession](#) initialisiert.

HRESULT Logout ()

Beschreibung: Die Methode ermöglicht das Abmelden der Komponente vom enaio® Applikations - Server.

Rückgabewerte:

IResult

Hinweis: mögliche Fehlercodes

0 - kein Fehler

Beispiel:

```
DrtSession.Logout
```

Siehe: [Allgemeine Beispiele](#) , [IDrtSession::Login](#)

HRESULT Connected ([out, retval] VARIANT_BOOL * *pbConnected*)

Beschreibung: Diese Methode gibt an, ob eine Verbindung zum enaio® Applikations - Server besteht.

Rückgabewerte:

*VARIANT_BOOL** *pbConnected*: True = Verbindung besteht, ansonsten False

Beispiel:

```
VARIANT_BOOL*pbConnected = DrtSession.Connected
```

Siehe: [Allgemeine Beispiele](#)

HRESULT UserName ([out, retval] BSTR * *pbstrUserName*)

Beschreibung: Diese Methode gibt den Namen des angemeldeten Benutzers an.

Rückgabewerte:

*BSTR** *pbstrUserName*: Name des Benutzers

Beispiel:

```
BSTR* pbstrUserName = DrtSession.UserName
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ServerName ([out, retval] BSTR * *pbstrServerName*)

Beschreibung: Diese Methode liefert den Namen des enaio® Applikations - Servers.

Rückgabewerte:

*BSTR** *pbstrServerName*: Servername

Beispiel:

```
BSTR* pbstrServerName = DrtSession.ServerName
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ServerPort ([out, retval] BSTR * *pbstrServerPort*)

Beschreibung: Diese Methode gibt den Port des enaio® Applikations - Servers an.

Rückgabewerte:

*BSTR**pbstrServerPort: Serverport

Beispiel:

```
BSTR* pbstrServerPort = DrtSession.ServerPort
```

Siehe: [Allgemeine Beispiele](#)

HRESULT AppName ([out, retval] BSTR * *pbstrAppName*)

Beschreibung: Diese Methode liefert den Namen der angemeldeten enaio® Applikation.

Rückgabewerte:

*BSTR**pbstrAppName: Applikationsname

Beispiel:

```
BSTR* pbstrAppName = DrtSession.AppName
```

Siehe: [Allgemeine Beispiele](#)

HRESULT License ([out, retval] BSTR * *pbstrLicense*)

Beschreibung: Diese Methode liefert den Namen der verwendeten Lizenz.

Rückgabewerte:

*BSTR**pbstrLicense: Name der Lizenz

Beispiel:

```
BSTR* pbstrLicense = DrtSession.License
```

Siehe: [Allgemeine Beispiele](#)

HRESULT hWnd ([out, retval] long * *phWnd*)

Beschreibung: Diese Methode liefert die Speicheradresse des Dialogfensters.

Rückgabewerte:

*LONG**phWnd: Speicheradresse des verwendeten Dialogfensters

Beispiel:

```
LONG* phWnd = DrtSession.hWnd
```

Siehe: [Allgemeine Beispiele](#)

HRESULT DbCursorCount ([out, retval] long * *plCount*)

Beschreibung: Diese Methode liefert die Anzahl der Datenbankzeiger. Standardmäßig ist der Wert auf 1 gesetzt.

Rückgabewerte:

*LONG**plCount: Anzahl der Datenbankzeiger

Beispiel:

```
LONG* plCount = DrtSession.DbCursorCount
```

Siehe: [Allgemeine Beispiele](#)

HRESULT GetObjDefXml ([out, retval] BSTR * *pbstrXml*)

Beschreibung: Diese Methode dient zur Anfrage nach der Objektdefinition.

Rückgabewerte:

*BSTR**pbstrXML: String, der die Objektdefinition in XML Form enthält

Beispiel:

```
BSTR* pbstrXML = DrtSession.GetObjDefXml
```

Siehe: [Allgemeine Beispiele](#)

HRESULT GetResultFromXml ([in] BSTR *bstrXml*, [out, retval] [IDrtResults](#) ** *ppResults*)

Beschreibung: Diese Methode liefert eine Anfrage nach einer Trefferliste im XML Format.

Parameter:

BSTR bstrXml: String, der die Anfrage des Servers in XML Form enthält

Rückgabewerte:

*IDrtResults**ppResults* **Hinweis:** mögliche Fehlercodes

0 - kein Fehler

16 -

Beispiel:

```
IDrtResults**ppResults = DrtSession.GetResultFromXml (BSTR bstrXML)
```

Siehe: [Allgemeine Beispiele](#) , [GetResult](#) , [IDrtResults](#)

HRESULT GetResult ([in] [IDrtQueries](#) * *pQueries*, [out, retval] [IDrtResults](#) ** *ppResults*)

Beschreibung: Anfrage im einfachen Datenbeschreibungsformat nach einer Zusammenstellung von DrtQuery Objekten.

Parameter:

*IDrtQueries** pQueries: Anfrage-Objekt

Rückgabewerte:

*IDrtResults*** ppResults

Beispiel:

```
IDrtResults**ppResults = DrtSession.GetResult (IDrtQueries*pQueries)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT GetResultObjectsFromXml ([in] BSTR *bstrXml*, [out, retval] [IDrtObjects](#) ** *ppResultObjects*)

Beschreibung: Diese Methode liefert eine Anfrage nach einer Trefferliste im XML Format.

Parameter:

BSTR bstrXml: String, der die Anfrage des Servers in XML Form enthält

Rückgabewerte:

*IDrtObjects** ppResultObjects*: Ergebnisliste in Form einer Zusammenstellung vom Typ [IDrtObjects](#)

Beispiel:

```
IDrtObjects**ppResultObjects = DrtSession.GetResultObjectsFromXml (BSTR
bstrXML)
```

Siehe: [Allgemeine Beispiele](#) , [GetResultObjects](#) , [IDrtObjects](#)

HRESULT GetResultObjects ([in] [IDrtQueries](#) * *pQueries*, [out, retval] [IDrtObjects](#) ** *ppResultObjects*)

Beschreibung: Anfragenach eine Objektdatenbeschreibung von der DrtQuery Objekten Zusammenstellung.

Parameter:

IDrtQueries pQueries*: Anfrage-Objekt, das die Suchbedingungen enthält

Rückgabewerte:

*IDrtObjects** ppObjects*

Beispiel:

```
IDrtObjects**ppObjects = DrtSession.GetResultObjects (IDrtQueries*pQueries)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT InsertFromXml ([in] BSTR *bstrXml*, [in] DrtImportFlagEnum *Flags*, [out, retval] long * *pObjectID*)

Beschreibung: Diese Methode vollzieht das Einfügen eines Indexdatenobjektes in den Datenbestand des enaio® Applikations - Servers. Als Parameter dient hier ein XML String.

Parameter:

BSTR bstrXml: String der die Serialisierung eines einzelnen Objektes in XML Form enthält
DrtImportFlagEnum Flags

Rückgabewerte:

LONG pObjectID*: ID des angelegten Objektes in der Datenbank

Beispiel:

```
LONG* pObjectID = DrtSession.InsertFromXml (BSTR bstrXml, DrtImportFlagEnum
Flags)
```

Siehe: [Allgemeine Beispiele](#) , [Insert](#)

HRESULT Insert ([in, out] [IDrtObject](#) * *pObject*, [in] DrtImportFlagEnum *Flags*, [out, retval] long * *pObjectID*)

Beschreibung: Diese Methode vollzieht das Einfügen eines Indexdatenobjektes in den Datenbestand des enaio® Applikations - Servers. Als Parameter dient hier ein Objekt vom Typ [IDrtObject](#).

Parameter:

*IDrtObject** pObj: Zeiger auf ein einzelnes Objekt vom Typ [IDrtObject](#)
DrtImportFlagEnum Flags

Rückgabewerte:

*LONG** plObjectID: ID des angelegten Objektes in der Datenbank

Beispiel:

```
LONG* plObjectID = DrtSession.Insert(IDrtObject*pObject,DrtImportFlagEnum Flags)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT UpdateFromXml ([in] BSTR *bstrXml*, [in] DrtImportFlagEnum *Flags*)

Beschreibung: Diese Methode vollzieht ein Update eines vorhandenen Indexdatenobjektes auf dem enaio® Applikations - Servers. Als Parameter dient hier ein XML String.

Parameter:

BSTR bstrXml: String der die Serialisierung eines einzelnen Objektes in XML Form enthält
DrtImportFlagEnum Flags:

Beispiel:

```
DrtSession.UpdateFromXml(BSTR bstrXml,DrtImportFlagEnum Flags)
```

Siehe: [Allgemeine Beispiele](#) , [Update](#)

HRESULT Update ([in] [IDrtObject](#) * *pObject*, [in] DrtImportFlagEnum *Flags*)

Beschreibung: Diese Methode vollzieht ein Update eines vorhandenen Indexdatenobjektes auf dem enaio® Applikations - Servers. Als Parameter dient hier ein Objekt vom Typ [IDrtObject](#).

Parameter:

*IDrtObject** pObj: Zeiger auf ein einzelnes Objekt vom Typ [IDrtObject](#)
DrtImportFlagEnum Flags:

Beispiel:

```
DrtSession.Update(IDrtObject*pObject,DrtImportFlagEnum Flags)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT DeleteFromXml ([in] BSTR *bstrXml*, [in] DrtImportFlagEnum *Flags*)

Beschreibung: Diese Methode vollzieht ein Löschen eines vorhandenen Indexdatenobjektes auf dem enaio® Applikations - Servers. Als Parameter dienen hier:

Parameter:

BSTR bstrXml
DrtImportFlagEnum Flags

Beispiel:

```
DrtSession.DeleteFromXml(BSTR bstrXml,DrtImportFlagEnum Flags)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Delete ([in] [IDrtObject](#) * *pObject*, [in] DrtImportFlagEnum *Flags*)

Beschreibung: Diese Methode vollzieht ein Löschen eines vorhandenen Indexdatenobjektes auf dem enaio® Applikations - Servers. Als Parameter dienen hier:

Parameter:

*IDrtObject***pObject*
DrtImportFlagEnum *Flags*

Beispiel:

```
DrtSession.Delete(IDrtObject*pObject,DrtImportFlagEnum Flags)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT MoveFromXml ([in] BSTR *bstrXml*, [in] DrtImportFlagEnum *Flags*)

Beschreibung: Diese Methode verschiebt ein vorhandenes Indexdatenobjektes auf dem enaio® Applikations-Server an einen neuen Standort. Die neue Folder-ID und die neue Register-ID müssen im übergebenen Objekt eingetragen sein. Als Parameter dienen hier:

Parameter:

BSTR *bstrXml*
DrtImportFlagEnum *Flags*

Beispiel:

```
DrtSession.MoveFromXml(BSTR bstrXml,DrtImportFlagEnum Flags)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Move ([in] [IDrtObject](#) * *pObject*, [in] DrtImportFlagEnum *Flags*)

Beschreibung: Diese Methode verschiebt ein vorhandenes Indexdatenobjektes auf dem enaio® Applikations - Server an einen neuen Standort. Die neue Folder-ID und die neue Register-ID müssen im übergebenen Objekt eingetragen sein. Als Parameter dienen hier:

Parameter:

*IDrtObject** *pObject*
DrtImportFlagEnum *Flags*

Beispiel:

```
DrtSession.Move(IDrtObject*pObject,DrtImportFlagEnum Flags)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT CreateRelationMgr ([out, retval] [IDrtRelationMgr](#) ** *ppRelationMgr*)

Beschreibung: Diese Methode erstellt ein Relationsmanagerobjekt, welches man benutzt um das Relationstextobjekt und deren Liste von Serverjobs zu überwachen.

Rückgabewerte:

*IDrtRelationMgr*** *ppRelationMgr*: Zeiger auf den Zeiger der Schnittstelle [IDrtRelationMgr](#)

Beispiel:

```
IDrtRelationMgr**ppRelationMgr = DrtSession.CreateRelationMgr
```

Siehe: [Allgemeine Beispiele](#) , [IDrtRelationMgr](#)

HRESULT CreatePortfolioMgr ([out, retval] [IDrtPortfolioMgr](#) ** ppPortfolioMgr)

Beschreibung: Diese Methode erstellt ein Portfoliomanagerobjekt, welches man benutzt um Portfolio und deren Liste von Serverjobs zu überwachen.

Rückgabewerte:

*IDrtPortfolioMgr*** ppPortfolioMgr: Zeiger auf den Zeiger der Schnittstelle [IDrtPortfolioMgr](#)

Beispiel:

```
IDrtPortfolioMgr**ppPortfolioMgr = DrtSession.CreatePortfolioMgr
```

Siehe: [Allgemeine Beispiele](#) , [IDrtPortfolioMgr](#)

HRESULT CreateAccessObj ([out, retval] [IDrtAccess](#) ** ppAccess)

Beschreibung: Diese Methode erstellt ein Zugriffsobjekt, welches man benutzt um Zugriffe und deren Liste von Serverjobs zu überwachen.

Rückgabewerte:

*IDrtAccess*** ppAccess: Zeiger auf den Zeiger der Schnittstelle [IDrtAccess](#)

Beispiel:

```
IDrtAccess**ppAccess = DrtSession.CreateAccessObj
```

Siehe: [Allgemeine Beispiele](#) , [IDrtAccess](#)

HRESULT CreateSecurityObj ([out, retval] [IDrtSecurity](#) ** ppSecurity)

Beschreibung: Diese Methode erstellt ein Sicherheitsobjekt, welches man benutzt um Benutzer oder Gruppeninformation zu erhalten.

Rückgabewerte:

*IDrtSecurity*** ppSecurity: Zeiger auf den Zeiger der Schnittstelle [IDrtSecurity](#)

Beispiel:

```
IDrtSecurity**ppSecurity = DrtSession.CreateSecurityObj
```

Siehe: [Allgemeine Beispiele](#) , [IDrtSecurity](#)

HRESULT CreateUserMgr ([out, retval] [IDrtUserMgr](#) ** ppUserMgr)

Beschreibung: Diese Methode erstellt ein Benutzerverwaltungsobjekt.

Rückgabewerte:

*IDrtUserMgr*** ppUserMgr: Zeiger auf den Zeiger der Schnittstelle [IDrtUserMgr](#)

Beispiel:

```
IDrtUserMgr**ppUserMgr = DrtSession.CreateUserMgr
```

Siehe: [Allgemeine Beispiele](#)

HRESULT CreateJob ([in] BSTR *bsNamespace*, [in] BSTR *bsFuncname*, [out, retval] IDrtJob ** *ppJob*)

Beschreibung: Diese Methode erstellt ein Jobobject zum Ausführen allgemeiner Serverjobs.

Rückgabewerte:

*IDrtJob**ppJob*: Zeiger auf den Zeiger der Schnittstelle IDrtJob

Beispiel:

```
IDrtJob*ppJob = DrtSession.CreateJob
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ResetWorkTime ()

Beschreibung: Diese Methode setzt den Arbeitszeitähler auf Null zurück.

Beispiel:

```
DrtSession.ResetWorkTime
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ResetJobTime ()

Beschreibung: Diese Methode setzt den Jobzeitähler auf Null zurück.

Beispiel:

```
DrtSession.ResetJobTime
```

Siehe: [Allgemeine Beispiele](#)

HRESULT WorkTime ([out, retval] long * *plTime*)

Beschreibung: Diese Methode gibt die benötigte Arbeitszeit in Millisekunden zurück.

Rückgabewerte:

*LONG*plTime*: Zeitangabe in Millisekunden

Beispiel:

```
LONG* plTime = DrtSession.WorkTime
```

Siehe: [Allgemeine Beispiele](#)

HRESULT JobTime ([out, retval] long * *plTime*)

Beschreibung: Diese Methode gibt die benötigte Jobzeit in Millisekunden zurück.

Rückgabewerte:

*LONG*plTime*: Zeitangabe in Millisekunden

Beispiel:

```
LONG* plTime = DrtSession.JobTime
```

Siehe: [Allgemeine Beispiele](#)

HRESULT LoadObjectFiles ([in] [IDrtObject](#) * *pObject*, [in] DrtLoadObjFileFlagsEnum *enFlags*, [out, retval] BSTR * *pbstrFiles*)

Beschreibung: Diese Methode lädt die Dokument-Dateien zum gegebenen DRT Objekt vom Server.

Parameter:

*IDrtObject** *pObject* gibt das Objekt an, für das die Dateien vom Server geladen werden sollen
DrtImportFlagEnum Flags für das Laden der Dateien

Rückgabewerte:

*BSTR** *pbstrFiles* Komma-Separierte Liste der Dokumentdateien

Beispiel:

```
BSTR strFileNames = DrtSession.LoadObjectFiles(pObject, 0)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT SaveObjectFiles ([in] [IDrtObject](#) * *pObject*, [in] DrtSaveObjFileFlagsEnum *enFlags*, [in] BSTR *bstrFiles*, [out, retval] long * *plSize*)

Beschreibung: Diese Methode speichert die Dokument-Dateien zum gegebenen DRT Objekt auf dem Server.

Parameter:

*IDrtObject** *pObject* gibt das Objekt an, für das die Dateien auf dem Server gespeichert werden sollen
DrtImportFlagEnum Flags für das Speichern der Dateien
BSTR *strFileNames* Komma-Separierte Liste der Dokumentdateien

Beispiel:

```
DrtSession.SaveObjectFiles(pObject, 0, strFileNames)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT LastError ([out, retval] BSTR * *pVal*)

Beschreibung: Diese Eigenschaft liefert einen Fehlertext für den letzten in der Session aufgetretenen Fehler.

Parameter:

*BSTR** *pbstrError* gibt eine kurze Beschreibung des innerhalb der Session zuletzt aufgetretenen Fehlers an

Beispiel:

```
strError = DrtSession.LastError
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ErrorDetails ([out, retval] BSTR * *pVal*)

Beschreibung: Diese Eigenschaft liefert eine Fehlerbeschreibung für den letzten in der Session aufgetretenen Fehler.

Parameter:

*BSTR** *pbstrDetails* gibt eine detaillierte Beschreibung des letzten Fehlers an

Beispiel:

```
strDetails = DrtSession.ErrorDetails
```

Siehe: [Allgemeine Beispiele](#)

HRESULT LastHits ([out, retval] long * *pVal*)

Beschreibung: Diese Eigenschaft liefert die Anzahl der Treffer der letzten Anfrage

Parameter:

*LONG** pLastHits gibt die Trefferanzahl der letzten Anfrage an (Unterscheidet sich bei von der Anzahl der Objekt in den Ergebnislisten, wenn Seitenweise Anfragen verwendet werden.)

Beispiel:

```
lastHits = DrtSession.LastHits
```

Siehe: [Allgemeine Beispiele](#)

HRESULT UserID ([out, retval] long * *pVal*)

Beschreibung: Diese Eigenschaft liefert die ID des angemeldeten Benutzers

Parameter:

*LONG** pVal gibt ID des angemeldeten Benutzers zurück

Beispiel:

```
lMyID = DrtSession.UserID
```

Siehe: [Allgemeine Beispiele](#)

HRESULT OSLocalWorkDir ([out, retval] BSTR * *bstrPath*)

Beschreibung: Diese Methode liefert den Pfad zum OS WorkDir zurück.

Rückgabewerte:

*BSTR** pbstrWorkDirPath Komma-Separierte Liste der Dokumentdateien

Beispiel:

```
WorkDir = DrtSession.OSLocalWorkDir
```

Siehe: [Allgemeine Beispiele](#)

HRESULT GetCurrentUserData ([in] BSTR *bstrName*, [in] long *lType*, [out, retval] BSTR * *bstrData*)

Beschreibung: Diese Methode Benutzerdaten für den aktuellen Benutzer entsprechend dem angefragten Datentyp.

Parameter:

LONG lType gibt den Typ der Benutzerdaten an

Rückgabewerte:

*BSTR** pbstrUserData liefert die Benutzerdaten als String zurück

Beispiel:

```
Userdata = DrtSession.GetUserData (type)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ShowLoginDlg ([out, retval] VARIANT_BOOL * *pVal*)

Beschreibung: Mit Hilfe dieser Eigenschaft kann gesteuert werden ob ein Login Dialog bei der Anmeldung am System angezeigt wird. Wird diese Eigenschaft gesetzt kann über die Anmeldung von Benutzername DIALOG ohne Passwort ein Anmeldedialog erzeugt werden. Im Falle einer fehlgeschlagenen Antwort erscheint ebenfalls ein Anmeldedialog.

Rückgabewerte:

*LONG***pVal* Wert, der das Flag enthält

Beispiel:

```
ShowLoginDialog = DrtSession.IsShowLoginDlg
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ShowLoginDlg ([in] VARIANT_BOOL *newVal*)

Beschreibung: Mit Hilfe dieser Eigenschaft kann gesteuert werden ob ein Login Dialog bei der Anmeldung am System angezeigt wird. Wird diese Eigenschaft gesetzt kann es dazu führen das Objekte bei einer Anfrage nicht gefunden werden.

Parameter:

*LONG**pVal* Wert, der das Flag enthält

Beispiel:

```
DrtSession.IsShowLoginDlg = ShowLoginDialog
```

Siehe: [Allgemeine Beispiele](#)

HRESULT CheckLicense ([in] BSTR *bsLicense*, [out, retval] long * *lRet*)

Beschreibung: Mit Hilfe dieser Funktion wird ermittelt, ob die angefragte Lizenz auf dem aktuell verfügbar ist. Wird ein Resultwert ungleich 0 zurückgegeben, so ist diese Lizenz nicht verfügbar.

Parameter:

BSTR *bsLicense* Lizenzstring (die einzelnen Lizenzen sind durch Leerzeichen voneinander getrennt)

Rückgabewerte:

*LONG**lRet* 0 wenn die angefragten Lizenzen verfügbar sind. Sonst ungleich 0.

Beispiel:

```
ret = DrtSession.CheckLicense(strLicense)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT GetDeletedObjects ([in, optional, defaultvalue(0)] long *lUserId*, [in, optional, defaultvalue(NULL)] [IDrtQueries](#) * *pQueries*, [out, retval] [IDrtObjects](#) ** *ppResultObjects*)

Beschreibung: Anfrage nach eine Objektdatenbeschreibung der Objekte, die im Papierkorb des angefragten Benutzers liegen. Wird die Benutzer ID mit 0 angegeben, so wird der Papierkorb des aktuell angemeldeten Benutzers ausgelesen. Bei einer Benutzer ID von -1 wird der Systempapierkorb ausgelesen. Hat der Benutzer nicht das Recht den angefragten Papierkorb zu lesen so wird ein Fehler zurückgegeben. Analog zu andern Anfragen kann über eine Anfragequery im Papierkorb recherchiert werden. Die Trefferliste des Papierkorbes kann ebenso wie andere Anfragen über entsprechende Parameter in der Query seitenweise angefragt werden.

Parameter:

LONGIUserId: Id des angefragter Benutzers (Default: 0 (aktueller Benutzer), -1 (Systempapierkorb))
*IDrtQueries**: Anfrageobjekt, das die Anfragebedingungen enthält (Default: 0 (alle Objekttypen werden angefragt))
LONGIPagesize: Anzahl der pro Anfrage zu ermittelnden Treffer (Default: -1)

Rückgabewerte:

*IDrtObjects*** ppObjects

Beispiel:

```
Dim objects As DrtObjects
Dim queries As DrtQueries
Dim user As DrtUser
Set user = GetSomeUser
... ' build the queries object
Set objects = DrtSession.GetDeletedObjects(user.Id, queries)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT RestoreObject ([in] [IDrtObject](#) * pObject, [in, optional, defaultValue(DrtRestore_Single)] DrtRestoreFlagsEnum flags)

Beschreibung: Ein im Papierkorb ermitteltes Objekt kann wieder hergestellt werden. Ist ein Object beispielsweise ein Register, so können alle in diesem Register enthaltenen Dokumente ebenfalls mithergestellt werden.

Parameter:

*IDrtObject** pObject: Objekt, das wiederhergestellt werden soll
DrtRestoreFlagsEnum flag: Es soll nur das aktuell gegebene Objekt wiederhergestellt werden, oder es sollen alle von diesem Objekt abhängigen Objekte (z.B. Dokumente im Register) ebenfalls wiederhergestellt werden DrtRestore_Single (0, default) = nur das aktuelle Objekt wiederherstellen
DrtRestore_All (1) = alle abhängigen Objekte ebenfalls wiederherstellen

Beispiel:

```
Dim object As DrtObject
Set object = FindMyDeletedObject
DrtSession.RestoreObject(object, DrtRestore_Single)
```

Siehe: [Allgemeine Beispiele](#)

IDrtTableField Schnittstellenreferenz

Abgeleitet von IDispatch.

Ausführliche Beschreibung

Diese Schnittstelle beschreibt ein spezielles Feldobjekt: Tabellenfeld.

Öffentliche Methoden

HRESULT [GetColCount](#) ([out, retval] long *plCount)
 HRESULT [GetColName](#) ([in] long lColIndex, [out, retval] BSTR *pbstrName)
 HRESULT [AddCol](#) ([in] BSTR bstrColName, [out, retval] long *plColIndex)
 HRESULT [GetRowCount](#) ([out, retval] long *plCount)
 HRESULT [AddRow](#) ([out, retval] long *plRowIndex)
 HRESULT [GetEntry](#) ([in] long lRowIndex, [in] VARIANT varCol, [out, retval] BSTR *pbstrValue)
 HRESULT [SetEntry](#) ([in] long lRowIndex, [in] VARIANT varCol, [in] BSTR bstrValue)
 HRESULT [Clear](#) (void)
 HRESULT [DeleteRow](#) (long lRowIndex)

Dokumentation der Elementfunktionen

HRESULT GetColCount ([out, retval] long *plCount)

Beschreibung: Diese Methode erhält die Anzahl der Spalten eines Tabellenfeldes zurück.

Rückgabewerte:

*LONG**plCount: Anzahl der Spalte eines Tabellenfeldes

Beispiel:

```
lCount = DrtTableField.GetColCount
```

Siehe: [Allgemeine Beispiele](#)

HRESULT GetColName ([in] long lColIndex, [out, retval] BSTR *pbstrName)

Beschreibung: Diese Methode erhält den Namen einer Spalte eines Tabellenfeldes.

Parameter:

*LONG*lColIndex: Index der Spalte.

Rückgabewerte:

*BSTR**bstrName: Spaltenname.

Beispiel:

```
bstrName = DrtTableField.GetColName(lColIndex)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT AddCol ([in] BSTR bstrColName, [out, retval] long *plColIndex)

Beschreibung: Diese Methode fügt eine neue Spalte in einem Tabellenfeld hinzu.

Parameter:

BSTR bstrName: Spaltenname.

Rückgabewerte:

*LONG** plColIndex: Index der neu angelegten Spalte.

Beispiel:

```
lColIndex = DrtTableField.AddCol(bstrName)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT GetRowCount ([out, retval] long * *plCount*)

Beschreibung: Diese Methode erhält die Anzahl der Ergebniszeilen eines Tabellenfeldes.

Rückgabewerte:

*LONG** plCount: Zeilenzahl

Beispiel:

```
lCount = DrtTableField.GetRowCount
```

Siehe: [Allgemeine Beispiele](#)

HRESULT AddRow ([out, retval] long * *plRowIndex*)

Beschreibung: Diese Methode erzeugt eine neue Zeile in einem Tabellenfeld und gibt den neuen Zeilenindex zurück.

Rückgabewerte:

*LONG** plRowIndex: Index der neu angelegten Zeile.

Beispiel:

```
lRowIndex = DrtTableField.AddRow
```

Siehe: [Allgemeine Beispiele](#)

HRESULT GetEntry ([in] long *lRowIndex*, [in] VARIANT *varCol*, [out, retval] BSTR * *pbstrValue*)

Beschreibung: Diese Methode gibt einen einzelnen Tabelleneintrag aus dem Tabellenfeld zurück. Die Spalte des Tabelleneintrages kann dabei wahlweise über den Namen als auch über den numerischen Spaltenindex angegeben werden.

Parameter:

LONG lRowIndex: Index der Zeile.

VARIANT varCol: Index oder Name der Spalte

Rückgabewerte:

*BSTR** pbstrValue: Wert des Tabelleneintrages.

Beispiel:

```
bstrValue = DrtTableField.GetEntry(lRowIndex, varCol)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT SetEntry ([in] long *lRowIndex*, [in] VARIANT *varCol*, [in] BSTR *bstrValue*)

Beschreibung: Diese Methode erzeugt einen neuen Tabelleneintrag in Tabellenfeld mit spezifischen Angaben über die Position (Zeile / Spalte). Die Spalte des Tabelleneintrages kann dabei wahlweise über den Namen als auch über den numerischen Spaltenindex angegeben werden.

Parameter:

*LONG*lRowIndex: Index der Zeile.

VARIANT varCol: Index der Spalte.

BSTR bstrValue: String des Tabelleneintrages.

Beispiel:

```
DrtTableField.SetEntry(lRowIndex, varCol, bstrValue)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Clear (void)

Beschreibung: Diese Methode entfernt alle Einträge des Tabellenfeldes (alle Zeilen und Spalten).

Beispiel:

```
DrtTableField.Clear
```

Siehe: [Allgemeine Beispiele](#)

HRESULT DeleteRow (long *lRowIndex*)

Beschreibung: Diese Methode entfernt einen einzelnen Eintrag, der durch den Index gegeben ist aus dem Tabellenfeld.

Parameter:

*LONG*lRowIndex: Index von einem Tabellenfeld

Beispiel:

```
DrtTableField.DeleteRow (lRowId)
```

Siehe: [Allgemeine Beispiele](#)

IDrtUser Schnittstellenreferenz

Abgeleitet von IDispatch.

Ausführliche Beschreibung

Diese Schnittstelle bietet den Zugriff auf die Daten eines Benutzers im System.

Öffentliche Methoden

HRESULT [Deserialize](#) ([in] BSTR bsXml)
HRESULT [Serialize](#) ([out, retval] BSTR *pbsXml)
HRESULT [AccountType](#) ([out, retval] long *pVal)
HRESULT [AccountType](#) ([in] long newVal)
HRESULT [Flags](#) ([out, retval] long *pVal)
HRESULT [Flags](#) ([in] long newVal)
HRESULT [Geaendert](#) ([out, retval] long *pVal)
HRESULT [Geaendert](#) ([in] long newVal)
HRESULT [LangID](#) ([out, retval] long *pVal)
HRESULT [LangID](#) ([in] long newVal)
HRESULT [ID](#) ([out, retval] long *pVal)
HRESULT [ID](#) ([in] long newVal)
HRESULT [Benutzer](#) ([out, retval] BSTR *pVal)
HRESULT [Benutzer](#) ([in] BSTR newVal)
HRESULT [Bemerkung](#) ([out, retval] BSTR *pVal)
HRESULT [Bemerkung](#) ([in] BSTR newVal)
HRESULT [LoginCount](#) ([out, retval] long *pVal)
HRESULT [LoginCount](#) ([in] long newVal)
HRESULT [LoginTime](#) ([out, retval] long *pVal)
HRESULT [LoginTime](#) ([in] long newVal)
HRESULT [LoginStation](#) ([out, retval] BSTR *pVal)
HRESULT [LoginStation](#) ([in] BSTR newVal)
HRESULT [Locked](#) ([out, retval] long *pVal)
HRESULT [Locked](#) ([in] long newVal)
HRESULT [Email](#) ([out, retval] BSTR *pVal)
HRESULT [Email](#) ([in] BSTR newVal)
HRESULT [GUID](#) ([out, retval] BSTR *pVal)
HRESULT [GUID](#) ([in] BSTR newVal)
HRESULT [Name](#) ([out, retval] BSTR *pVal)
HRESULT [Name](#) ([in] BSTR newVal)
HRESULT [Passwort](#) ([out, retval] BSTR *pVal)
HRESULT [Passwort](#) ([in] BSTR newVal)
HRESULT [Profil](#) ([out, retval] long *pVal)
HRESULT [Profil](#) ([in] long newVal)
HRESULT [ServerID](#) ([out, retval] long *pVal)
HRESULT [ServerID](#) ([in] long newVal)
HRESULT [Station](#) ([out, retval] BSTR *pVal)
HRESULT [Station](#) ([in] BSTR newVal)
HRESULT [Supervisor](#) ([out, retval] long *pVal)
HRESULT [Supervisor](#) ([in] long newVal)
HRESULT [ValidFrom](#) ([out, retval] BSTR *pVal)
HRESULT [ValidFrom](#) ([in] BSTR newVal)
HRESULT [ValidTo](#) ([out, retval] BSTR *pVal)
HRESULT [ValidTo](#) ([in] BSTR newVal)

Dokumentation der Elementfunktionen

HRESULT Deserialize ([in] BSTR *bsXml*)

Beschreibung: Diese Methode initialisiert ein [IDrtUser](#) - Objekt einschliesslich seiner Attribute und Unterobjekte aus einem XML String.

Parameter:

BSTR bsXml: xml-input string

Beispiel:

```
IDrtUser.Deserialize(bsXml)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Serialize ([out, retval] BSTR **pbsXml*)

Beschreibung: Diese Methode serialisiert das [IDrtUser](#) - Objekt einschliesslich seiner Attribute und Unterobjekte vollständig in einen XML String.

Rückgabewerte:

*BSTR**pbsXml: output xml-string

Beispiel:

```
pbsXml = IDrtUser.Serialize
```

Siehe: [Allgemeine Beispiele](#)

HRESULT AccountType ([out, retval] long **pVal*)

Beschreibung: Eigenschaft definiert den Benutzer account_type.

Rückgabewerte:

*LONG**pVal: Eigenschaft den account_type

Beispiel:

```
pVal = IDrtUser.AccountType
```

Siehe: [Allgemeine Beispiele](#)

HRESULT AccountType ([in] long *newVal*)

Beschreibung: Eigenschaft definiert den Benutzer account_type.

Parameter:

*LONG*newVal: Eigenschaft den account_type

Beispiel:

```
IDrtUser.AccountType(newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Flags ([out, retval] long **pVal*)

Beschreibung Diese Eigenschaft definiert in einem Flag Eigenschaften des Benutzers.

Rückgabewerte:*LONG**pVal**Beispiel:**

```
pVal = IDrtUser.Flags
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Flags ([in] long *newVal*)

Beschreibung: Diese Eigenschaft definiert in einem Flag Eigenschaften des Benutzers.

Parameter:*LONG*newVal**Beispiel:**

```
IDrtUser.Flags(newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Geaendert ([out, retval] long * *pVal*)

Beschreibung: Diese Eigenschaft definiert das Datum und die Uhrzeit der letzten Änderung der Benutzereigenschaften.

Rückgabewerte:*LONG**pVal Datum der letzten Änderung**Beispiel:**

```
pVal = IDrtUser.Geaendert
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Geaendert ([in] long *newVal*)

Beschreibung: Diese Eigenschaft definiert das Datum und die Uhrzeit der letzten Änderung der Benutzereigenschaften.

Parameter:*LONG*newVal Datum der letzten Änderung**Beispiel:**

```
IDrtUser.Geaendert(newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT LangID ([out, retval] long * *pVal*)

Beschreibung: Diese Eigenschaft erhält die Benutzer Sprachen ID.

Rückgabewerte:*LONG**pVal Benutzer Sprach ID**Beispiel:**

```
pVal = IDrtUser.LangID
```

Siehe: [Allgemeine Beispiele](#)

HRESULT LangID ([in] long *newVal*)

Beschreibung: Diese Eigenschaft setzt die Benutzer Sprachen ID.

Parameter:

*LONG*newVal Benutzer Sprach ID

Beispiel:

```
IDrtUser.LangID (newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ID ([out, retval] long * *pVal*)

Beschreibung: Diese Eigenschaft definiert die Datenbank-ID eines Benutzers.

Rückgabewerte:

*LONG**pVal Datenbank ID des Benutzers

Beispiel:

```
pVal = IDrtUser.ID
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ID ([in] long *newVal*)

Beschreibung: Diese Eigenschaft definiert die Datenbank-ID eines Benutzers.

Parameter:

*LONG*newVal Datenbank ID des Benutzers

Beispiel:

```
IDrtUser.ID (newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Benutzer ([out, retval] BSTR * *pVal*)

Beschreibung: Diese Eigenschaft definiert den Anmeldenamen eines Benutzers.

Rückgabewerte:

*BSTR**pVal: AnmeldeName

Beispiel:

```
pVal = IDrtUser.Benutzer
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Benutzer ([in] BSTR *newVal*)

Beschreibung: Diese Eigenschaft definiert den Anmeldenamen eines Benutzers.

Parameter:

BSTR newVal: Anmeldenamen:

Beispiel:

```
IDrtUser.Benutzer(newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Bemerkung ([out, retval] BSTR * *pVal*)

Beschreibung: Diese Eigenschaft definiert einen Beschreibungstext zum aktuellen Benutzer.

Rückgabewerte:

*BSTR**pVal: Beschreibungstext

Beispiel:

```
pVal = IDrtUser.Bemerkung
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Bemerkung ([in] BSTR *newVal*)

Beschreibung: Diese Eigenschaft definiert einen Beschreibungstext zum aktuellen Benutzer.

Parameter:

BSTR newVal: Beschreibungstext

Beispiel:

```
IDrtUser.Bemerkung(newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT LoginCount ([out, retval] long * *pVal*)

Beschreibung: Die Eigenschaft erhält die Benutzer Loginzählung.

Rückgabewerte:

*LONG**pVal: Loginzähler

Beispiel:

```
pVal = IDrtUser.LoginCount
```

Siehe: [Allgemeine Beispiele](#)

HRESULT LoginCount ([in] long *newVal*)

Beschreibung: Die Eigenschaft setzt die Benutzer Loginzählung.

Parameter:

LONG newVal: Loginzähler

Beispiel:

```
IDrtUser.LoginCount(newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT LoginTime ([out, retval] long * *pVal*)

Beschreibung: Diese Eigenschaft erhält die Benutzer Einlog- Zeit und - Datum.

Rückgabewerte:

*LONG** *pVal*: Einlog- Zeit -Datum

Beispiel:

```
pVal = IDrtUser.LoginTime
```

Siehe: [Allgemeine Beispiele](#)

HRESULT LoginTime ([in] long *newVal*)

Beschreibung: Diese Eigenschaft setzt die Benutzer Einlog- Zeit und -Datum.

Parameter:

LONG *newVal*: Einlog- Zeit -Datum

Beispiel:

```
IDrtUser.LoginTime(newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT LoginStation ([out, retval] BSTR * *pVal*)

Beschreibung: Diese Eigenschaft erhält die Login Station vom Benutzer.

Rückgabewerte:

*BSTR** *pVal*: Login Station vom Benutzer

Beispiel:

```
pVal = IDrtUser.LoginStation
```

Siehe: [Allgemeine Beispiele](#)

HRESULT LoginStation ([in] BSTR *newVal*)

Beschreibung: Diese Eigenschaft setzt die Login Station vom Benutzer.

Parameter:

BSTR *newVal*: Login Station vom Benutzer

Beispiel:

```
IDrtUser.LoginStation(newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Locked ([out, retval] long * *pVal*)

Beschreibung: Diese Eigenschaft erhält ob der Benutzer gesperrt ist oder nicht.

Rückgabewerte:

*LONG***pVal*: Ist Gesperrt?

Beispiel:

```
pVal = IDrtUser.Locked
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Locked ([in] long *newVal*)

Beschreibung: Diese Eigenschaft setzt den Benutzer auf gesperrt.

Parameter:

*LONG**newVal*

Beispiel:

```
IDrtUser.Locked(newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Email ([out, retval] BSTR * *pVal*)

Beschreibung: Diese Eigenschaft erhält die Benutzer Email Adresse.

Rückgabewerte:

*BSTR***pVal*: Benutzer Email Adresse

Beispiel:

```
pVal = IDrtUser.Email
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Email ([in] BSTR *newVal*)

Beschreibung: Diese Eigenschaft setzt die Benutzer Email Adresse.

Parameter:

BSTR *newVal*: Benutzer Email Adresse

Beispiel:

```
IDrtUser.Email(newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT GUID ([out, retval] BSTR * *pVal*)

Beschreibung: Diese Eigenschaft erhält die GUID von dem identifizierten Benutzer.

Rückgabewerte:

*BSTR***pVal*: GUID

Beispiel:

```
pVal = IDrtUser.GUID
```

Siehe: [Allgemeine Beispiele](#)

HRESULT GUID ([in] BSTR *newVal*)

Beschreibung: Diese Eigenschaft setzt die GUID für einen zu identifizierenden Benutzer.

Parameter:

BSTR newVal: GUID

Beispiel:

```
IDrtUser.GUID(newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Name ([out, retval] BSTR * *pVal*)

Beschreibung: Diese Eigenschaft erhält den Namen vom identifizierten Benutzer.

Rückgabewerte:

*BSTR**pVal: Name des Benutzers

Beispiel:

```
pVal = IDrtUser.Name
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Name ([in] BSTR *newVal*)

Beschreibung: Diese Eigenschaft setzt den Namen für den identifizierten Benutzer.

Parameter:

BSTR newVal: Name des Benutzers

Beispiel:

```
IDrtUser.Name(newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Passwort ([out, retval] BSTR * *pVal*)

Beschreibung: Diese Eigenschaft erhält das Benutzerpassword.

Rückgabewerte:

*BSTR**pVal: Benutzerpassword

Beispiel:

```
pVal = IDrtUser.Passwort
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Passwort ([in] BSTR *newVal*)

Beschreibung: Diese Eigenschaft setzt das Benutzerpassword.

Parameter:

BSTR *newVal*: Benutzerpassword

Beispiel:

```
IDrtUser.Passwort(newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Profil ([out, retval] long * *pVal*)

Beschreibung: Diese Eigenschaft definiert die ID des Benutzerprofils.

Rückgabewerte:

*LONG** *pVal*: ID des Benutzerprofils

Beispiel:

```
pVal = IDrtUser.Profil
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Profil ([in] long *newVal*)

Beschreibung: Diese Eigenschaft definiert die ID des Benutzerprofils.

Parameter:

LONG *newVal*: ID des Benutzerprofils

Beispiel:

```
IDrtUser.Profil(newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ServerID ([out, retval] long * *pVal*)

Beschreibung: Diese Eigenschaft erhält die Benutzer Server ID.

Rückgabewerte:

*LONG** *pVal*: Benutzer Server ID

Beispiel:

```
pVal = IDrtUser.ServerID
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ServerID ([in] long *newVal*)

Beschreibung: Diese Eigenschaft setzt die Benutzer Server ID.

Parameter:

LONG *newVal*: Benutzer Server ID

Beispiel:

```
IDrtUser.ServerID(newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Station ([out, retval] BSTR **pVal*)

Beschreibung: Diese Eigenschaft erhält den Benutzer Stationsname.

Rückgabewerte:

*BSTR***pVal*: Benutzer Stationsname

Beispiel:

```
pVal = IDrtUser.Station
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Station ([in] BSTR *newVal*)

Beschreibung: Diese Eigenschaft setzt den Benutzer Stationsanme.

Parameter:

BSTR *newVal*: Benutzer Stationsname

Beispiel:

```
IDrtUser.Station(newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Supervisor ([out, retval] long **pVal*)

Beschreibung: Diese Eigenschaft erhält den Wert ob der Benutzer Supervisor Rechte besitzt.

Rückgabewerte:

*LONG***pVal*: Benutzerrecht Supervisor

Beispiel:

```
pVal = IDrtUser.Supervisor
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Supervisor ([in] long *newVal*)

Beschreibung: Diese Eigenschaft setzt den Wert ob der Benutzer Supervisor Rechte besitzt.

Parameter:

LONG *newVal*: Benutzerrecht Supervisor

Beispiel:

```
IDrtUser.Supervisor(newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ValidFrom ([out, retval] BSTR * *pVal*)

Beschreibung: Diese Eigenschaft gibt an von wann bis wann der Benutzer gültig ist, sich also am System anmelden kann.

Rückgabewerte:

*BSTR***pVal*: Anfangsdatum der Gültigkeit

Beispiel:

```
pVal = IDrtUser.ValidFrom
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ValidFrom ([in] BSTR *newVal*)

Beschreibung: Diese Eigenschaft gibt an von wann bis wann der Benutzer gültig ist, sich also am System anmelden kann.

Parameter:

BSTR *newVal*: Anfangsdatum der Gültigkeit

Beispiel:

```
IDrtUser.ValidFrom(newVal)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ValidTo ([out, retval] BSTR * *pVal*)

Beschreibung: Diese Eigenschaft gibt an von wann bis wann der Benutzer gültig ist, sich also am System anmelden kann.

Rückgabewerte:

*BSTR***pVal*: Enddatum der Gültigkeit

Beispiel:

```
pVal = IDrtUser.ValidTo
```

Siehe: [Allgemeine Beispiele](#)

HRESULT ValidTo ([in] BSTR *newVal*)

Beschreibung: Diese Eigenschaft gibt an von wann bis wann der Benutzer gültig ist, sich also am System anmelden kann.

Parameter:

BSTR *newVal*: Enddatum der Gültigkeit

Beispiel:

```
IDrtUser.ValidTo(newVal)
```

Siehe: [Allgemeine Beispiele](#)

IDrtUserMgr Schnittstellenreferenz

Abgeleitet von IDispatch.

Ausführliche Beschreibung

Der Manager bietet Möglichkeiten zum Erstellen, Bearbeiten und Löschen von Benutzern im System.

Öffentliche Methoden

HRESULT [CreateUser](#) ([in] [IDrtUser](#) *newUser,[out, retval] long *lResult)
 HRESULT [UpdateUser](#) ([in] long updFlags,[in] [IDrtUser](#) *pUser,[in] [IDrtGroups](#) *pGroups,[out, retval] long *lResult)
 HRESULT [DeleteUser](#) ([in] [IDrtUser](#) *pUser,[in] [IDrtUser](#) *pTarget,[in] long InheritFlags,[out, retval] long *lResult)
 HRESULT [AddUsersToGroups](#) ([in] [IDrtUsers](#) *pUsers,[in] [IDrtGroups](#) *pGroups,[out, retval] long *lResult)
 HRESULT [AddGroupMember](#) ([in] [IDrtUser](#) *pUser,[in] [IDrtGroup](#) *pGroup,[out, retval] long *lResult)
 HRESULT [RemoveGroupMember](#) ([in] [IDrtUser](#) *pUser,[in] [IDrtGroup](#) *pGroup,[out, retval] long *lResult)
 HRESULT [CreateGroup](#) ([in] [IDrtGroup](#) *newGroup,[out, retval] long *lResult)
 HRESULT [EmptyGroup](#) ([in] [IDrtGroup](#) *pGroup,[out, retval] long *lResult)
 HRESULT [UpdateGroup](#) ([in] long updFlags,[in] [IDrtGroup](#) *pGroup,[in] [IDrtUsers](#) *pUsers,[out, retval] long *lResult)
 HRESULT [DeleteGroup](#) ([in] [IDrtGroup](#) *pGroup,[out, retval] long *lResult)

Dokumentation der Elementfunktionen

HRESULT CreateUser ([in] [IDrtUser](#) *newUser, [out, retval] long *lResult)

Beschreibung: Diese Methode erstellt mit Hilfe des Aufrufs mng.CreateUser Job einen neuen Benutzer. Die Benutzerdefinitionen bekommt diese Methode über die Schnittstelle [IDrtUser](#).

Parameter:

*IDrtUser**newUser: Neuer Benutzer der erstellt werden soll. ID und GUID werden ignoriert und durch die Datenübernahme von der Schnittstelle [IDrtUser](#) verändert.

Rückgabewerte:

*LONG**lResult: Rückgabewert ob Benutzer angelegt ist.

Beispiel:

```
LONG* lResult = IDrtUserMgr.CreateUser(IDrtUser*newUser)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT UpdateUser ([in] long updFlags, [in] [IDrtUser](#) *pUser, [in] [IDrtGroups](#) *pGroups, [out, retval] long *lResult)

Beschreibung: Diese Methode setzt (modifiziert) Benutzerattribute und \ oder Gruppenverbände beim Aufruf von mng.SetUserAttributes Job. Die Benutzerdefinitionen bekommt diese Methode über die Schnittstelle [IDrtUser](#).

Parameter:

*LONG*updFlags: Flag was zeigt welches Feld durch die Daten von [IDrtUser](#) benutzt wird für das Update.

*IDrtUser**pUser: Benutzer der modifiziert werden soll. Jede kleinste Eigenschaft Benutzer, ID or GUID muss gültig für den Benutzer sein, ist eine der Eigenschaften Benutzer oder GUID gültig für einen Anderen so wird dies leer gelassen. Sind Benutzer und GUID leer so wird die ID Eigenschaft als Input für den Job genutzt.

*IDrtGroups**pGroups: 0 oder Gruppenliste zu welche man den Benutzer als Mitglied hinzufügen möchte.

Rückgabewerte:

*LONG**lResult: Rückgabewert ob Benutzer verändert wurde

Beispiel:

```
LONG* lResult = IDrtUserMgr.UpdateUser(LONG updFlags, IDrtUser*pUser,
IDrtGroups*pGroups)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT DeleteUser ([in] [IDrtUser](#) *pUser, [in] [IDrtUser](#) *pTarget, [in] long InheritFlags, [out, retval] long *lResult)

Beschreibung: Diese Methode löscht beim Aufruf von mng.DeleteUser Job den Benutzer.

Parameter:

*IDrtUser**pUser: Benutzer, der zu löschen ist. Die Eigenschaften wie Benutzername, BenutzerID oder GUID geben den Benutzer eindeutig an. Sind Benutzername und GUID leer, so wird die ID Eigenschaft als Input für den Job genutzt.

pTarget: Benutzer, dem die existierenden Archiv-Objekte und Mappen des gelöschten Benutzers zugeordnet werden, damit sie im System nicht verloren gehen.

InheritFlags: 0 - Mappen, Archiv-Objekte werden keinem neuen Benutzer zugeordnet 1 - Mappen werden dem Benutzer pTarget zugeordnet 2 - Archiv-Objekte werden dem Benutzer pTarget zugeordnet 3 - Mappen und Archiv-Objekte werden dem Benutzer pTarget zugeordnet

Rückgabewerte:

*LONG**lResult: Rückgabewert, der den Erfolg der Löschaktion beschreibt

*/** Deletes user calling mng.DeleteUser job.*

Parameter:

pUser - user to be deleted. One of Properties Benutzer, ID or GUID of user should correspond to valid user. If Benutzer and GUID are empty the ID property will be used as input of the job.

pTarget - Target user. If input parameter pUser has nonzero Benutzer property, then pTarget should have nonzero Benutzer property, that correspond to valid user, and the same for ID and GUID.

InheritFlags - Flags for Job. If InheritFlags = 0 - don't hand over maps and mail to Target, 1 - hand over maps to Target, 2 - hand over mail to Target, 3 (1 & 2) - hand over maps and mail to Target.

Rückgabewerte:

lResult - returned error result.

HRESULT AddUsersToGroups ([in] [IDrtUsers](#) *pUsers, [in] [IDrtGroups](#) *pGroups, [out, retval] long *lResult)

Beschreibung: Diese Methode fügt Benutzern (oder einem Benutzer sofern die Liste nur einen Benutzer enthält) zu spezifischen Gruppen oder Gruppe hinzu durch Aufruf von mng.AddUserGroupAsc Job.

Parameter:

*IDrtUsers**pUsers: Liste von Benutzern die hinzu gefügt werden sollen. GUID der Benutzer müssen eindeutig sein.

*IDrtGroups**pGroups: Liste von Gruppen zu welchen man die Benutzer hinzufügen möchte. GUID der Gruppen muss eindeutig sein.

Rückgabewerte:

*LONG**lResult: Rückgabewert ob Benutzer zu den Gruppen hinzugefügt wurden

Beispiel:

```
LONG* lResult =  
IDrtUserMgr.AddUsersToGroups (IDrtUsers*pUsers, IDrtGroups*pGroups)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT AddGroupMember ([in] [IDrtUser](#) *pUser, [in] [IDrtGroup](#) *pGroup, [out, retval] long *lResult)

Beschreibung: Diese Methode fügt Benutzer zu einer spezifischen Gruppe durch Aufruf von mng.AddUserGroupAsc Job hinzu.

Parameter:

*IDrtUser**pUser: Benutzer der hinzugefügt werden soll. GUID vom Benutzer muss eindeutig sein.

*IDrtGroup**pGroup: Gruppe zu welcher der Benutzer hinzugefügt werden soll. GUID der Gruppe muss eindeutig sein.

Rückgabewerte:

*LONG**lResult: Rückgabewert ob Benutzer zu der Gruppe hinzugefügt wurde

Beispiel:

```
LONG* lResult = IDrtUserMgr.AddGroupMember (IDrtUser*pUser, IDrtGroup*pGroup)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT RemoveGroupMember ([in] [IDrtUser](#) *pUser, [in] [IDrtGroup](#) *pGroup, [out, retval] long *lResult)

Beschreibung: Diese Methode entfernt Benutzer von spezifischen Gruppen durch Aufruf des mng.AddUserGroupAsc Job.

Parameter:

*IDrtUser**pUser: Benutzer der entfernt werden soll. GUID vom Benutzer muss eindeutig sein.

*IDrtGroup**pGroup: Gruppe von welcher der Benutzer entfernt werden soll. GUID der Gruppe muss eindeutig sein.

Rückgabewerte:

*LONG**lResult: Rückgabewert ob Benutzer von der Gruppe entfernt wurde.

Beispiel:

```
LONG* lResult =  
IDrtUserMgr.RemoveGroupMember (IDrtUser*pUser, IDrtGroup*pGroup)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT CreateGroup ([in] [IDrtGroup](#) *newGroup, [out, retval] long *lResult)

Beschreibung: Diese Methode erstellt eine Gruppe durch Aufruf des mng.CreateGroup Job. Die Gruppendefinition bekommt diese Methode über die Schnittstelle [IDrtGroup](#).

Parameter:

*IDrtGroup**newGroup: Neue Gruppe die erstellt werden soll. ID und GUID der Gruppe wird ignoriert und durch die Eingabe der Schnittstelle [IDrtGroup](#) verändert.

Rückgabewerte:

*LONG**lResult: Rückgabewert ob Gruppe erstellt wurde.

Beispiel:

```
LONG* lResult = IDrtUserMgr.CreateGroup (IDrtGroup*newGroup)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT EmptyGroup ([in] [IDrtGroup](#) * pGroup, [out, retval] long * lResult)

Beschreibung: Diese Methode entfernt alle Mitglieder von der spezifischen Gruppe.

Parameter:

*IDrtGroup**pGroup: Gruppe von welcher die Benutzer geleerten werden soll. GUID der Gruppe muss eindeutig sein.

Rückgabewerte:

*LONG**lResult: Rückgabewert ob Gruppe geleert wurde.

Beispiel:

```
LONG* lResult = IDrtUserMgr.EmptyGroup (IDrtGroup*pGroup)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT UpdateGroup ([in] long updFlags, [in] [IDrtGroup](#) * pGroup, [in] [IDrtUsers](#) * pUsers, [out, retval] long * lResult)

Beschreibung: Diese Methode setzt oder modifiziert Gruppenattribute und \ oder Mitglieder durch Aufruf des mng.SetGroupAttributes Jobs. Die Gruppeneffinition bekommt diese Methode über die Schnittstelle [IDrtGroup](#).

Parameter:

*LONG*updFlags: Flag zeigt welches Feld für das Update durch [IDrtGroup](#) benutzt werden soll.

*IDrtGroup**pGroup: Gruppe welche modifiziert werden soll.

pGroup - group to be modified. At least one of properties Name, ID or GUID should be of valid group, if one of properties Name or GUID is valid other ones of them could be empty. If Name and GUID are empty the ID property will be used as input of the job.

pUsers - NULL or list of users which will be added to the group as members.

Rückgabewerte:

lResult - returned error result.

HRESULT DeleteGroup ([in] [IDrtGroup](#) * pGroup, [out, retval] long * lResult)

Deletes group calling mng.DeleteGroup job.

Parameter:

pGroup - group to be deleted. One of Properties Name, ID or GUID of group should correspond to valid group. If Name and GUID are empty the ID property will be used as input of the job.

Rückgabewerte:

lResult - returned error result.

IDrtUsers Schnittstellenreferenz

Abgeleitet von IDispatch.

Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Liste von Objekten vom Typ [IDrtUser](#).

Siehe auch: [IDrtUser](#)

Öffentliche Methoden

HRESULT [Deserialize](#) ([in] BSTR bsXml)

HRESULT [Serialize](#) ([out, retval] BSTR *pbsXml)

Dokumentation der Elementfunktionen

HRESULT Deserialize ([in] BSTR *bsXml*)

Beschreibung: Diese Methode initialisiert ein [IDrtUsers](#) - Objekt einschliesslich seiner Attribute und Unterobjekte aus einem XML String.

Parameter:

BSTR bsXml: xml-input string

Beispiel:

```
IDrtUsers.Deserialize(bsXml)
```

Siehe: [Allgemeine Beispiele](#)

HRESULT Serialize ([out, retval] BSTR **pbsXml*)

Beschreibung: Diese Methode serialisiert das [IDrtUsers](#) - Objekt einschliesslich seiner Attribute und Unterobjekte vollständig in einen XML String.

Rückgabewerte:

*BSTR**pbsXml: output xml-string

Beispiel:

```
pbsXml = IDrtUsers.Serialize
```

Siehe: [Allgemeine Beispiele](#)