

*enaiio*<sup>®</sup>

# Softwaredokumentation COM-Objekte für Workflowevents

Version 8.50



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 – 2017 by  
OPTIMAL SYSTEMS GmbH  
Cicerostraße 26  
D-10709 Berlin

01.02.2017  
Version 8.50

# Inhalt

Einleitung	4
Zweck	4
Begriffserklärung	4
Aufbau der COM-Objekte für die enaio® - Workflow Engine	4
Standardobjekte	5
Aufbau der COM-Objekte für den enaio® - Workflow Client	7
Standardobjekte	7
Klassen-Verzeichnis	11
enaio® - Workflow Engine	11
enaio® - Workflow Client	13
Klassen Dokumentation	15
IActiveWorkItem Schnittstellenreferenz	15
IComActivity Schnittstellenreferenz	19
IComActivityList Schnittstellenreferenz	27
IComCollection Schnittstellenreferenz	28
IComDataField Schnittstellenreferenz	29
IComFile Schnittstellenreferenz	37
IComFileDoc Schnittstellenreferenz	39
IComFileDocs Schnittstellenreferenz	42
IComModel Schnittstellenreferenz	43
IComModelActivity Schnittstellenreferenz	46
IComModelActivityList Schnittstellenreferenz	49
IComOrgClass Schnittstellenreferenz	50
IComOrgClassList Schnittstellenreferenz	52
IComOrgManagement Schnittstellenreferenz	53
IComOrgObjAttribute Schnittstellenreferenz	55
IComOrgObjAttributeList Schnittstellenreferenz	57
IComOrgObject Schnittstellenreferenz	58
IComOrgObjectList Schnittstellenreferenz	66
IComPerformer Schnittstellenreferenz	67
IComPerformerList Schnittstellenreferenz	70
IComProcess Schnittstellenreferenz	71
IComTerminateActivityInfo Schnittstellenreferenz	82
IComTimer Schnittstellenreferenz	84
IComTimerFiredInfo Schnittstellenreferenz	95
IComTimers Schnittstellenreferenz	97
IComTimerActions Schnittstellenreferenz	98
IComEmail Schnittstellenreferenz	101
IComEmails Schnittstellenreferenz	105
IComEmailRecipient Schnittstellenreferenz	106
IComEmailRecipients Schnittstellenreferenz	108
IComSystem Schnittstellenreferenz	109
IFileObject Schnittstellenreferenz	112
IFileObjects Schnittstellenreferenz	118
IOrganisation Schnittstellenreferenz	119
IOrgAttribute Schnittstellenreferenz	124
IOrgAttributes Schnittstellenreferenz	126
IOrgClass Schnittstellenreferenz	127
IOrgClasses Schnittstellenreferenz	129
IOrgObject Schnittstellenreferenz	130
IOrgObjects Schnittstellenreferenz	136
IRoutingLists Schnittstellenreferenz	137
IRoutingList Schnittstellenreferenz	138
IRoutingListEntry Schnittstellenreferenz	143
IRoutingListItem Schnittstellenreferenz	144
IRoutingListItemParticipants Schnittstellenreferenz	151
IAdhocReminderPeriodTemplate Schnittstellenreferenz	152
IAdhocReminderPeriodTemplates Schnittstellenreferenz	154

IComRoutingList Schnittstellenreferenz.....	155
IComRoutingListEntry Schnittstellenreferenz .....	158
IComRoutingListItem Schnittstellenreferenz.....	160
IComAdhocReminderPeriodTemplates Schnittstellenreferenz .....	163
IComAdhocReminderPeriodTemplate Schnittstellenreferenz.....	164
IScriptResult Schnittstellenreferenz .....	165
IWFCClient Schnittstellenreferenz .....	168
IWFFieldValues Schnittstellenreferenz .....	170
IWFFile Schnittstellenreferenz.....	173
IWFListColumn Schnittstellenreferenz.....	176
IWFListColumns Schnittstellenreferenz .....	179
IWFMaskField Schnittstellenreferenz .....	180
IWFMaskFields Schnittstellenreferenz.....	185
IWFCtrlPage Schnittstellenreferenz .....	186
IWFCtrlPages Schnittstellenreferenz.....	187
IWFFVariable Schnittstellenreferenz.....	188
IWFFVariables Schnittstellenreferenz .....	197
Statusübersicht .....	198
Prozessstatus.....	198
Aktivitätsstatus .....	198
Codebeispiele .....	199
Email senden (Server-Event) .....	199
Neues Listenelement erstellen (Server-Event) .....	199
Bearbeiter einer Aktivität festlegen (Server-Event) .....	200
Prozessverantwortlichen löschen (Server-Event) .....	200
Auswahlliste an Maskentabellenfeldern dynamisch füllen (Client-Event) .....	201
Auswahlliste an Maskentextfeldern dynamisch füllen (Client-Event) .....	202
Verwendung von Serverjobs in Events.....	203
Index .....	206

# Einleitung

## Zweck

Dieses Dokument enthält die Dokumentation der COM-Objekte für die enaio® - Workflow Engine und den enaio® - Workflow Client. Alle in der Dokumentation enthaltenen Code-Beispiele sind für Client- bzw. Server-Events.

## Begriffserklärung

Workflowmodelle automatisieren Geschäftsvorfälle. Sie bestehen aus einer logischen Anordnung von Tätigkeiten (Aktivitäten), den zu bearbeitenden Daten (Workflowvariablen, Dokumente) und den Personen, welche die Tätigkeiten ausführen. Die Modellierung eines Workflowmodells erfolgt graphisch über den enaio® - Workflow Editor. Mit Hilfe eines Workflowmodells wird durch eine Clientanwendung ein Prozess erstellt, welcher eine konkrete Instanz des Modells bzw. Geschäftsvorfalls darstellt. Die Bearbeitung einzelner Aktivitäten eines Prozesses erfolgt über Clientanwendungen. Hierbei werden durch den jeweiligen Bearbeiter Daten manipuliert (Maskeneingabe, Dokumente). Der Prozessablauf wird durch die enaio® - Workflow Engine gesteuert, die Grundlage bildet hierbei das Workflowmodell. Über die Engine erfolgt auch die Zuordnung der Aktivitäten an die jeweiligen Bearbeiter. Während der Verarbeitung eines Prozesses kann mittels Events auf diesen Einfluss genommen werden. Dazu wird den Events Scriptcode (VB-Script) hinterlegt. In diesem kann nun über COM-Objekte und deren Methoden auf den Prozess zugegriffen werden. Events stehen sowohl in der enaio® - Workflow Engine, als auch in den Clientanwendungen zur Verfügung. Events können über den enaio® - Workflow Editor bearbeitet werden. Eine Liste der vorhandenen Events kann dem Editorhandbuch bzw. dem Workflow Systemhandbuch entnommen werden.

## Aufbau der COM-Objekte für die enaio® - Workflow Engine

Der Aufbau eines Workflowmodells, der enaio® - Workflow Editor definiert wurde, wird vollständig durch die COM-Schnittstellen abgebildet. Ein Workflowmodell wird durch die [IComModel](#) - Schnittstelle beschrieben. Wenn ein Workflowmodell gestartet wird, geht das Modell in einen Prozess über. Der Zugriff auf den Workflowprozess erfolgt über die [IComProcess](#) - Schnittstelle. Ein Prozess besteht aus mehreren Aktivitäten. Diese Aktivitäten sind Instanzen der Workflow-Aktivitäten. Bedingt durch Schleifen-durchläufe können innerhalb eines Prozesses, zu einer Aktivität, mehrere Instanzen erstellt werden. Um diese unterscheiden zu können und einen eindeutigen Zugriff auf Aktivitätenobjekte zu ermöglichen, wird für jede Aktivität eine Instanzid (auch RActivityId oder relative Id) vergeben, welche sich von der Id der Aktivität aus dem Workflow unterscheidet. Die zum Workflowmodell definierte Organisationsstruktur wird über die Schnittstelle [IComOrgManagement](#) abgebildet. Alle Organisationsklassen und Objekte sind über [IComOrgManagement](#) erreichbar. Alle Workflowvariablen werden mit der [IComDataField](#) - Schnittstelle bereitgestellt. Der Name eines Variablenobjektes entspricht dem Workflowvariablenamen.

## Standardobjekte

Diese Objekte stehen dem Benutzer immer zur Verfügung und müssen vom Benutzer nicht erstellt werden. Zu den Standardobjekten zählen:

- § **thisprocess** : Objekt der [IComProcess](#) - Schnittstelle
- § **wforganisation** : Objekt der [IComOrgManagement](#) - Schnittstelle
- § **wfsystem** : Objekt der [IComSystem](#) - Schnittstelle
- § **"workflowvariable"** : Objekte der [IComDataField](#) - Schnittstelle

### thisprocess

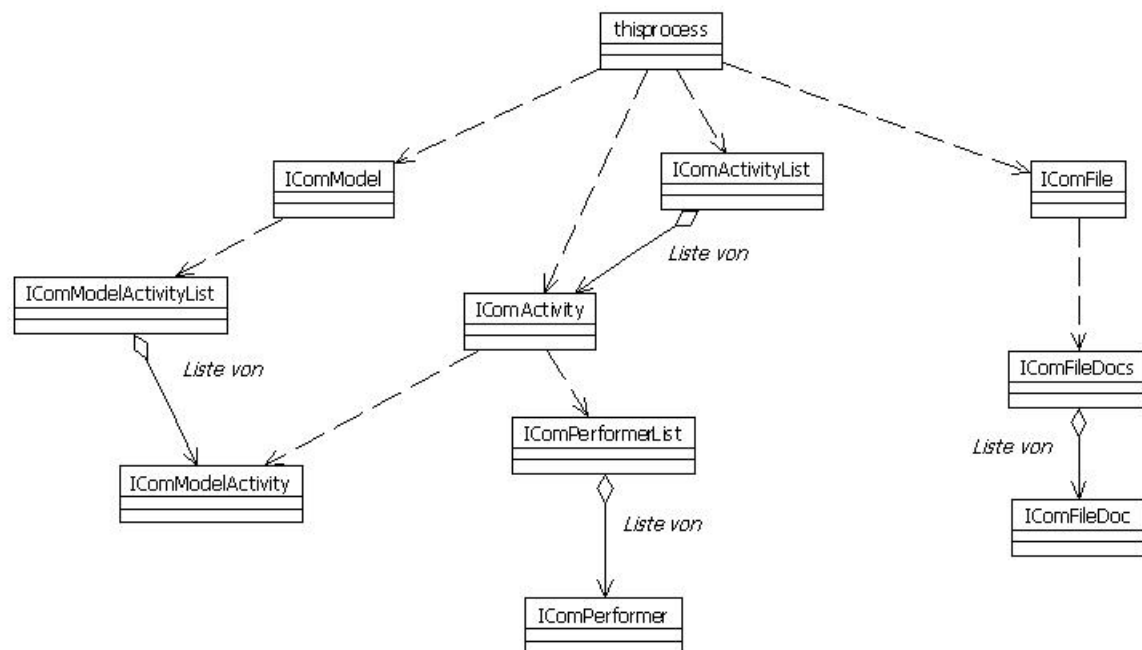
Das Objekt **thisprocess** bildet einen Workflowprozess ab. Ein Workflowprozess entsteht, wenn ein Workflowmodell im Client gestartet wird. Das Objekt liefert folgende Informationen zum Prozess:

- § Prozessid
- § Prozessname
- § Prozessersteller
- § Workflowname des Prozesses
- § Prozessverantwortliche
- §

Das Objekt stellt außerdem Methoden bereit über die Objekte folgender Schnittstellen erstellt werden:

- § [IComActivity](#): Bei Objekten dieser Schnittstelle handelt es sich um die Aktivitäten, die im Prozess erstellt wurden. Diese Aktivitäten sind Instanzen der Workflow-Aktivitäten.
- § [IComFile](#): Dieses Objekt stellt die Workflowakte dar. Über dieses Objekt kann auf den Akteninhalt zugegriffen werden.
- § [IComModel](#): Dieses Objekt liefert die Workflowmodell-Aktivitäten.
- § [IComTimer](#): Dieses Objekt repräsentiert eine Mahn- oder Sperrfrist auf dem Modell.

## Übersicht der über 'thisprocess' ansprechbaren Schnittstellen

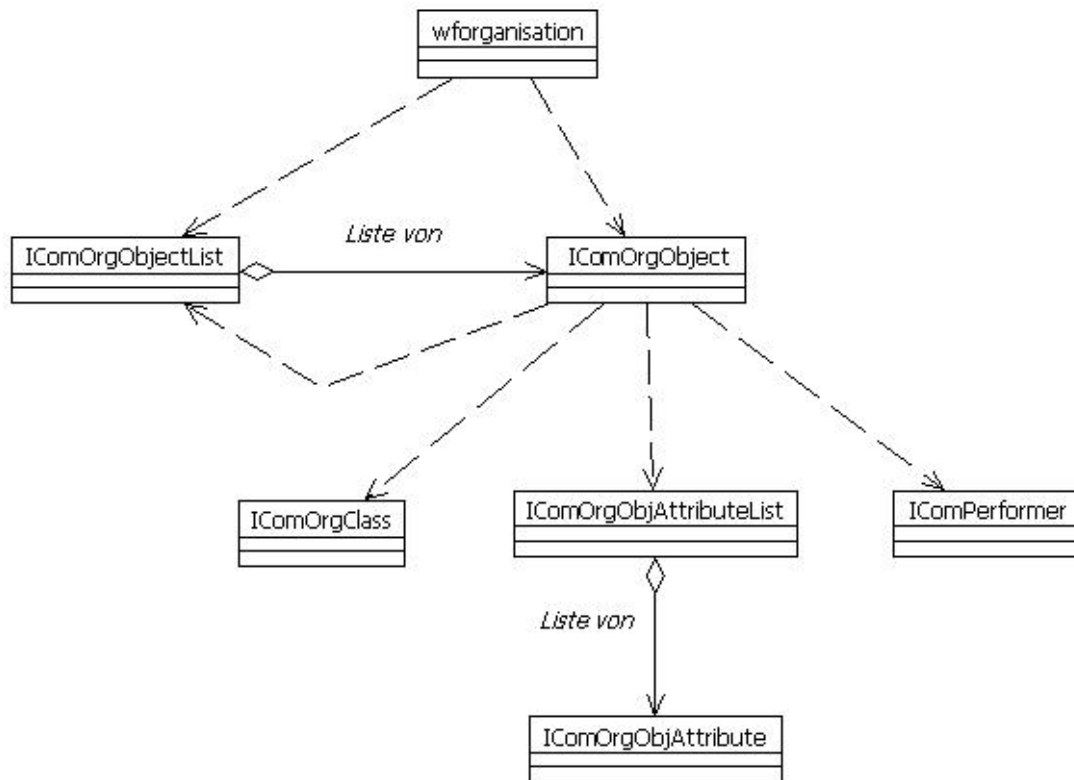


## wforganisation

Das Objekt **wforganisation** beschreibt die Organisationsverwaltung. Das Objekt stellt Methoden bereit, über die Objekte folgender Schnittstellen erstellt werden können:

- § [IComOrgObject](#): Bei Objekten dieser Schnittstelle handelt es sich um die Organisationsobjekte, der im enaio® - Workflow Editor definierten Organisationsklassen.
- § [IComOrgObjectList](#): Diese Schnittstelle beschreibt eine Liste von Organisationsobjekten.
- §

## Übersicht der über 'wforganisation' ansprechbaren Schnittstellen



### wfsystem

Das Objekt **wfsystem** stellt allgemeine Systemfunktionen bereit.

### "Name der Workflowvariable"

Das Objekt **"Name der Workflowvariable"** kapselt die im enaio® - Workflow Editor definierte Variable. Es wird für jede definierte Variable ein Objekt erstellt und ist über den Variablennamen aus dem enaio® - Workflow Editor ansprechbar. Das Objekt liefert folgende Informationen zur Variable:

- § Id
- § Typ
- § Wert
- § Vorbelegung
- § Prozessverantwortliche
- § Anzahl der Elemente in Listen
- §

Das Objekt stellt außerdem Methoden bereit, über die neue Listenobjekte erstellt werden können.

## Aufbau der COM-Objekte für den enaio® - Workflow Client

### Standardobjekte

Diese Objekte stehen dem Benutzer immer zur Verfügung und müssen vom Benutzer nicht erstellt werden. Zu den Standardobjekten zählen:

- § **wffile** : Objekt der [IWFFile](#) - Schnittstelle
- § **wforganisation** : Objekt der [IOrganisation](#) - Schnittstelle



- § **wfclient** : Objekt der [IWFCClient](#) - Schnittstelle
- § **scriptresult** : Objekt der [IScriptResult](#) - Schnittstelle
- § **workitem** : Objekt der [IActiveWorkItem](#) - Schnittstelle
- § **wfvariables** u. **wfprotocol** : Objekte der [IWFVariables](#) - Schnittstelle
- § **wfmaskfields** : Objekt der [IWFMaskFields](#) - Schnittstelle

## wffile

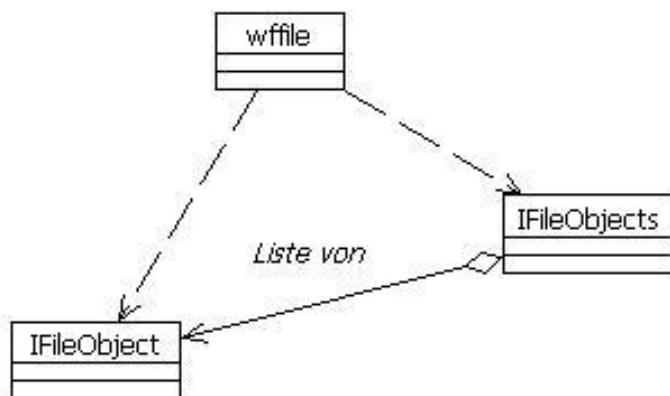
Das Objekt **wffile** bildet die Workflowakte ab. Das Objekt liefert folgende Informationen zur Akte:

- § Unzulässige Objekttypen
- §

Das Objekt stellt außerdem Methoden bereit, über die Objekte folgender Schnittstellen erstellt werden können:

- § [IFileObject](#): Bei Objekten dieser Schnittstelle handelt es sich um Dokumente, die sich in der Akte befinden.
- §

## Übersicht der über 'wffile' ansprechbaren Schnittstellen



## wforganisation

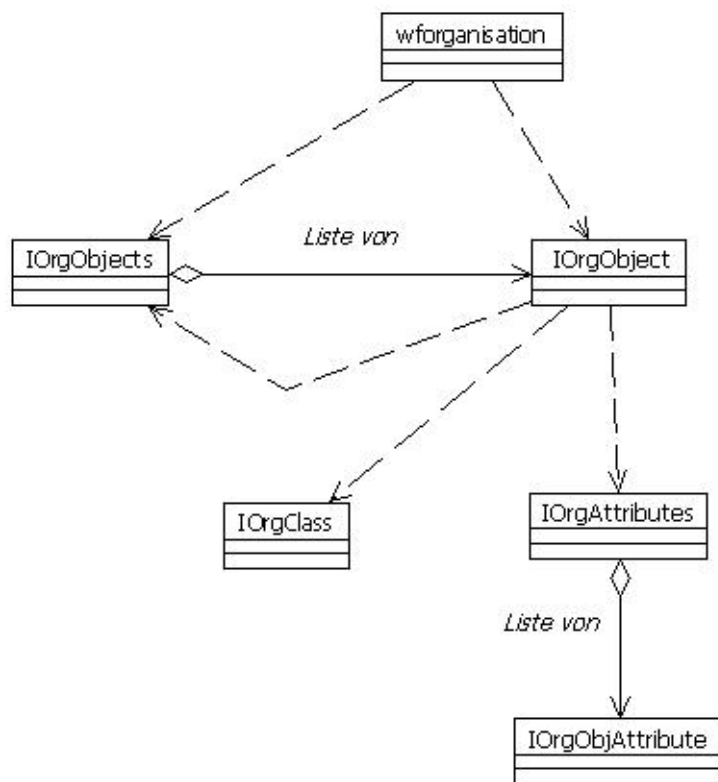
Das Objekt **wforganisation** bildet die Organisationsstruktur ab. Das Objekt liefert folgende Informationen zur Organisation:

- § Id der Organisation
- § Name
- § Status: aktiv/inaktiv
- §

Das Objekt stellt Methoden bereit, über die Objekte folgender Schnittstellen erstellt werden können:

- § [IOrgObject](#): Bei Objekten dieser Schnittstelle handelt es sich um die Organisationsobjekte, der im enaio® - Workflow Editor definierten Organisationsklassen.
- § [IOrgObjects](#): Diese Schnittstelle beschreibt eine Liste von Organisationsobjekten.
- §

### Übersicht der über 'wforganisation' ansprechbaren Schnittstellen



## wfclient

Das Objekt **wfclient** liefert Informationen zu einer Aktivität, wie z.B. die Teilnehmer der Aktivität und Abwesenheit eines Benutzers.

## scriptresult

Mit Hilfe dieses Objektes kann das Programmverhalten nach Ausführung eines VB-Scripts bestimmt werden.

## workitem

Das Objekt stellt den aktuell laufenden Vorgangsschritt dar. Mit Hilfe dieses Objektes kann eine Passwortabfrage von Benutzer vor dem Weiterleiten eines Vorgangsschritts erzwungen werden. Außerdem liefert dieses Objekt allgemeine Informationen zum Vorgangsschritt, wie z.B. Prozessid, Prozessname und Name der Aktivität.

### wfmaskfields

Das Objekt **wfmaskfields** ist eine Liste aller Maskenfelder. Die Eigenschaften eines Maskenfeldes, wie z.B. Sichtbarkeit und Schreibschutz, können verändert werden.

### wfvariables

Das Objekt **wfvariables** ist eine Liste aller Workflowvariablen. Das Objekt für die Protokollierung 'wfprotocol', ist auch ein Objekt dieser Schnittstelle.

# Klassen-Verzeichnis

## enaio® - Workflow Engine

Bereich	Klassen	Beschreibung
<b>Workflowmodell</b>	<a href="#"><u>IComModel</u></a>	Schnittstelle beschreibt ein Workflowmodell.
	<a href="#"><u>IComModelActivityList</u></a>	Liste von Modelaktivitäten.
	<a href="#"><u>IComModelActivity</u></a>	Schnittstelle liefert u.a. Id, Namen und Typ einer Aktivität aus dem Workflowmodell.
	<a href="#"><u>IComPerformerList</u></a>	Liste von Teilnehmers einer Aktivität.
	<a href="#"><u>IComPerformer</u></a>	Schnittstelle liefert u.a. Teilnehmer einer Aktivität.
<b>Organisationsstruktur</b>	<a href="#"><u>IComOrgManagement</u></a>	Schnittstelle liefert Organisationsobjekte anhand Id, Namen, Klassennamen und Personen + Rollen (in den Server-Events unter dem Namen 'wforganisation' verfügbar).
	<a href="#"><u>IComOrgClassList</u></a>	Liste von Organisationsklassen.
	<a href="#"><u>IComOrgClass</u></a>	Schnittstelle liefert Id und Namen der Organisationsklasse.
	<a href="#"><u>IComOrgObjectList</u></a>	Liste von Organisationsobjekten.
	<a href="#"><u>IComOrgObject</u></a>	Schnittstelle liefert u.a. Namen, direkten Vorgänger/Nachfolger, Attribute und Organisationsklasse des Organisationsobjektes.
	<a href="#"><u>IComOrgObjAttributeList</u></a>	Liste von Objektattributen.
	<a href="#"><u>IComOrgObjAttribute</u></a>	Schnittstelle liefert Id, Name und Wert eines Objektattributes.
<b>Prozess und Vorgangsschritte</b>	<a href="#"><u>IComProcess</u></a>	Schnittstelle liefert u.a. ProcessId, aktuelle Aktivität, Prozessersteller und Bearbeiter einer Aktivität kann gesetzt werden (in den Server-Events unter dem Namen 'thisprocess' verfügbar).
	<a href="#"><u>IComActivityList</u></a>	Liste von Aktivitäten.
	<a href="#"><u>IComActivity</u></a>	Schnittstelle liefert u.a. Namen, Bearbeiter, Workflowvariablen, Typ und Systemvariablen zur Aktivität.
	<a href="#"><u>IComDataField</u></a>	Schnittstelle ermöglicht Zugriff auf Workflowvariablen (in den Server-Events unter dem Variablennamen verfügbar).
	<a href="#"><u>IComPerformerList</u></a>	Liste von Teilnehmern einer Aktivität.

	<a href="#"><u>IComPerformer</u></a>	Schnittstelle liefert u.a. Teilnehmer einer Aktivität.
	<a href="#"><u>IComTimers</u></a>	Liste von Fristen
	<a href="#"><u>IComTimer</u></a>	Mahnfrist oder Sperrfrist
	<a href="#"><u>IComTimerActions</u></a>	Aktionen beim Ablauf einer Mahnfrist
	<a href="#"><u>IComEmails</u></a>	Liste von Emails, die bei Ablauf einer Mahnfrist verschickt werden
	<a href="#"><u>IComEmail</u></a>	Email, die bei Ablauf einer Mahnfrist verschickt werden
	<a href="#"><u>IComEmailRecipients</u></a>	Liste von Emailempfängern
	<a href="#"><u>IComEmailRecipient</u></a>	Emailempfänger
<b>Workflowakte und Dokumente</b>	<a href="#"><u>IComFile</u></a>	Schnittstelle liefert/löscht u.a. den Inhalt der Workflowakte.
	<a href="#"><u>IComFileDocs</u></a>	Liste aller Dokumente einer Akte.
	<a href="#"><u>IComFileDoc</u></a>	Schnittstelle liefert u.a. Typ und DokumentenId eines Dokumentes aus der Workflowakte.
<b>Sonstiges</b>	<a href="#"><u>IComSystem</u></a>	Schnittstelle stellt u.a. Methode zum Versenden einer E-Mail bereit (in den Server-Events unter dem Namen 'wfsystem' verfügbar).

Der Aufruf von Workflowjobs aus serverseitigen Workflowevents über den 'running context' ist nicht möglich. Hierbei kommt es zum Absturz von einaio® server.

## enaio® - Workflow Client

Bereich	Klassen	Beschreibung
<b>Organisationsstruktur</b>	<a href="#"><u>IOrganisation</u></a>	Schnittstelle beschreibt die Organisationsstruktur (in den Client-Events unter dem Namen 'wfororganisation' verfügbar).
	<a href="#"><u>IOrgObjects</u></a>	Liste von Organisationsobjekten.
	<a href="#"><u>IOrgObject</u></a>	Schnittstelle beschreibt ein Organisationsobjekt.
	<a href="#"><u>IOrgClasses</u></a>	Liste von Organisationsklassen
	<a href="#"><u>IOrgClass</u></a>	Schnittstelle beschreibt eine Organisationsklasse.
	<a href="#"><u>IOrgAttributes</u></a>	Liste von Objektattributen.
	<a href="#"><u>IOrgAttribute</u></a>	Schnittstelle beschreibt ein Objektattribut.
<b>Vorgangsschritt und Variablen</b>	<a href="#"><u>IActiveWorkItem</u></a>	Schnittstelle beschreibt personalisierten Vorgangsschritt (in den Client-Events unter dem Namen 'Workitem' verfügbar).
	<a href="#"><u>IWFVariables</u></a>	Liste aller Workflowvariablen (in den Client-Events unter dem Namen 'wfvariables' verfügbar).
	<a href="#"><u>IWFVariable</u></a>	Schnittstelle beschreibt eine Workflowvariable.
<b>Maskenfelder</b>	<a href="#"><u>IWFMaskFields</u></a>	Liste von Maskenfeldern (in den Client-Events unter dem Namen 'wfmaskfields' verfügbar).
	<a href="#"><u>IWFMaskField</u></a>	Schnittstelle liefert u.a. Eigenschaften eines Maskenfeldes.
	<a href="#"><u>IWFCtrlPages</u></a>	Liste von allen Seiten eines Pagecontrols
	<a href="#"><u>IWFCtrlPage</u></a>	Schnittstelle beschreibt eine Seite eines Pagecontrols
	<a href="#"><u>IWFFieldValues</u></a>	Schnittstelle beschreibt Auswahlliste an Textfeldern in Masken.
	<a href="#"><u>IWFListColumns</u></a>	Liste aller Spalten eines Maskenfeldes in Tabellenform.
	<a href="#"><u>IWFListColumn</u></a>	Schnittstelle beschreibt eine Spalte eines Maskenfeldes in Tabellenform.

<b>Workflowakte und Dokumente</b>	<a href="#"><u>IWFFile</u></a>	Schnittstelle beschreibt die Workflowakte (in den Client-Events unter dem Namen 'wffile' verfügbar).
	<a href="#"><u>IFileObjects</u></a>	Liste von Dokumenten der Workflowakte.
	<a href="#"><u>IFileObject</u></a>	Schnittstelle liefert Informationen zum Dokument aus dem enaio®.
<b>Sonstiges</b>	<a href="#"><u>IWFClient</u></a>	Schnittstelle ermöglicht Zugriff auf das Workflowsystem (in den Client-Events unter dem Namen 'wfclient' verfügbar).
	<a href="#"><u>IScriptResult</u></a>	Programmverhalten kann bestimmt werden (in den Client-Events unter dem Namen 'ScriptResult' verfügbar).

# Klassen Dokumentation

## IActiveWorkItem Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt einen personalisierten Vorgangsschritt (Aktivität). Eine personalisierte Aktivität ist einem Benutzer zugeordnet und kann nur noch von ihm bearbeitet werden. In Client-Events ist das Objekt unter dem Namen 'Workitem' verfügbar.

### Öffentliche Methoden

```
HRESULT WFMaskFields ([out, retval] IWFMaskFields **pVal)
HRESULT WFVariables ([out, retval] IWFFVariables **pVal)
HRESULT WFFile ([out, retval] IWFFFile **pVal)
HRESULT WorkflowID ([out, retval] BSTR *pVal)
HRESULT ProcessID ([out, retval] BSTR *pVal)
HRESULT ProcessName ([out, retval] BSTR *pVal)
HRESULT ActivityName ([out, retval] BSTR *pVal)
HRESULT ActivityID ([out, retval] BSTR *pVal)
HRESULT CheckPassword ([out, retval] VARIANT_BOOL *pVal)
HRESULT CheckPassword ([in] VARIANT_BOOL newVal)
HRESULT Save ([out] VARIANT *pvarErrorString,[out, retval]long *lpResult) HRESULT
RoutingList([out, retval] IRoutingList* **pVal);
HRESULT AdhocTemplateList([out,retval] IRoutingLists** ppRoutingLists)
HRESULT CheckAdhocTemplate([in] BSTR bstrTemplateId, [out, retval] VARIANT_BOOL*
pbIsValid)
HRESULT GetAdhocTemplate([in] BSTR bstrTemplateId, [out,retval] IRoutingList**
ppRoutingList)
```

### Dokumentation der Elementfunktionen

```
HRESULT WFMaskFields ([out, retval] IWFMaskFields ** pVal)
```

**Beschreibung:** Diese Methode liefert eine Liste aller Maskenfelder, der in der aktuellen Aktivität verwendeten Maske.

**Rückgabewerte:**

*IWFMaskFields:* Liste aller Maskenfelder

**Beispiel:**

```
Set MaskfieldList = Workitem.WFMaskFields
```



HRESULT WFVariables ([out, retval] [IWFVariables](#) \*\* pVal)

**Beschreibung:** Diese Methode liefert eine Liste aller Workflowvariablen.

**Rückgabewerte:**

*IWFVariables*: Liste der Workflowvariablen

**Beispiel:**

```
Set VariableList = Workitem.WFVariables
```

HRESULT WFFile ([out, retval] [IWFFile](#) \*\* pVal)

**Beschreibung:** Diese Methode liefert die Workflowakte eines Vorgangsschrittes.

**Rückgabewerte:**

*IWFFile*: Workflowakte

**Beispiel:**

```
Set WFFiles = Workitem.WFFile
```

HRESULT WorkflowID ([out, retval] BSTR \* pVal)

**Beschreibung:** Diese Methode liefert die ID des verwendeten Workflowmodells.

**Rückgabewerte:**

*BSTR*: Workflow-ID

**Beispiel:**

```
Workflow_ID = Workitem.WorkflowID
```

HRESULT ProcessID ([out, retval] BSTR \* pVal)

**Beschreibung:** Diese Methode liefert die Prozessid.

**Rückgabewerte:**

*BSTR*: Prozess-ID

**Beispiel:**

```
Process_ID = Workitem.ProcessID
```

HRESULT ProcessName ([out, retval] BSTR \* pVal)

**Beschreibung:** Diese Methode liefert die Prozessname.

**Rückgabewerte:**

*BSTR*: Prozessname

**Beispiel:**

```
Process_Name = Workitem.ProcessName
```

HRESULT ActivityName ([out, retval] BSTR \* *pVal*)

**Beschreibung:** Diese Methode liefert den Namen der Aktivität.

**Rückgabewerte:**

*BSTR*: Name der Aktivität

**Beispiel:**

```
Activity_Name = Workitem.ActivityName
```

HRESULT ActivityID ([out, retval] BSTR \* *pVal*)

**Beschreibung:** Diese Methode liefert die ID der Aktivität.

**Rückgabewerte:**

*BSTR*: ID der Aktivität

**Beispiel:**

```
Activity_ID = Workitem.ActivityID
```

HRESULT CheckPassword ([out, retval] VARIANT\_BOOL \* *pVal*)

**Beschreibung:** Diese Methode liefert den Status der Eigenschaft ‚CheckPassword‘. Ist die Eigenschaft auf TRUE gesetzt wird vom Benutzer vor dem Weiterleiten des Vorgangsschrittes eine Passwordeingabe erwartet.

**Rückgabewerte:**

*VARIANT\_BOOL*: TRUE = Passwortabfrage

**Beispiel:**

```
CheckPassword = Workitem.CheckPassword
```

HRESULT CheckPassword ([in] VARIANT\_BOOL *newVal*)

**Beschreibung:** Diese Methode setzt die Eigenschaft ‚CheckPassword‘. Ist die Eigenschaft auf TRUE gesetzt wird vom Benutzer vor dem Weiterleiten des Vorgangsschrittes eine Passwordeingabe erwartet.

**Parameter:**

*VARIANT\_BOOL*: TRUE = Passwortabfrage

**Beispiel:** Passwordeingabe vom Benutzer wird erwartet.

```
Workitem.CheckPassword = TRUE
```

HRESULT Save ([out] VARIANT \* *pvarErrorString*, [out, retval] long \* *lpResult*)

**Beschreibung:** Diese Methode speichert den aktuellen Zustand der Workflowvariablen und der Akte.

**Parameter:**

*VARIANT*: hier wird ggf. eine Fehlermeldung zurückgegeben

**Rückgabewerte:**

*long*: 0, wenn kein Fehler aufgetreten ist

**Beispiel:**

```
RetVal = Workitem.Save(ErrorString)
```

HRESULT RoutingList([out, retval] [IRoutingList](#)\* \*pVal);

**Beschreibung:** Diese Methode liefert die Abarbeitungsliste für einen Adhocarbeitsschritt. Bei Nicht-Adhocarbeitsschritten wird null zurückgeliefert.

**Rückgabewerte:**

[IRoutingList](#): die Abarbeitungsliste für einen Adhocarbeitsschritt. Bei Nicht-Adhocarbeitsschritten wird null zurückgeliefert.

**Beispiel:**

```
Set routinglist = Workitem.RoutingList
```

HRESULT AdhocTemplateList([out,retval] [IRoutingLists](#)\*\* ppRoutingLists)

**Beschreibung:** Diese Methode liefert alle dem angemeldeten Benutzer sichtbaren Adhoc-Routinglisten-Vorlagen.

**Rückgabewerte:**

[IRoutingLists](#): Liste mit Vorlagen

HRESULT CheckAdhocTemplate([in] BSTR bstrTemplateId, [out, retval] VARIANT\_BOOL\* pbIsValid)

**Beschreibung:** Diese Methode prüft, ob eine bestimmte Adhoc-Routinglisten-Vorlage für diesen Arbeitsschritt verwendet werden kann.

**Parameter:**

*BSTR*: bstrTemplateId – Id der Vorlage

**Rückgabewerte:**

VARIANT\_BOOL: gibt ab, ob die Vorlage für diesen Arbeitsschritt anwendbar ist

HRESULT GetAdhocTemplate([in] BSTR bstrTemplateId, [out,retval] [IRoutingList](#)\*\* ppRoutingList)

**Beschreibung:** Diese Methode liefert eine Lauflisten Vorlage anhand der Id.

**Parameter:**

*BSTR*: bstrTemplateId – Id der Vorlage

**Rückgabewerte:**

[IRoutingList](#)\*: die Vorlage

## IComActivity Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Aktivität. Bei diesen Objekten handelt es sich um die Aktivitäten, die im Prozess erstellt wurden. Also, um Instanzen von Aktivitäten aus dem Workflowmodell. Das bedeutet, dass zu einer Aktivität, die im Verlauf eines Prozesses mehrfach erstellt wurde (z.B. durch eine Schleife), auch mehrere Objekte existieren. Diese Objekte werden anhand der so genannten **InstanzID** (häufig auch RActivityId o. relative ID) unterschieden. **ActivityId** bezieht sich dagegen auf die Aktivitätenid aus dem Workflowmodell. Für den Zugriff auf ein Activity-Objekt in Server-Events gibt es folgende Möglichkeiten:

1. man erhält ein Objekt der aktuellen Aktivität

```
Set activity = thisprocess.CurrentActivity
```

2. man erhält ein Objekt einer beliebigen Aktivität über die ActivityID aus dem Workflowmodell

```
Dim MyActivityID 'ActivityID aus Workflowmodell  
  
Set activity = thisprocess.GetCurrentActivityById(MyActivityID)
```

3. man erhält ein Objekt einer beliebigen Aktivität über die InstanzId

```
Dim MyInstanceId 'InstanzId der Aktivität  
  
Set activity = thisprocess.GetRActivityById(MyInstanceId)
```

Siehe auch: [IComProcess](#)

## Öffentliche Methoden

HRESULT [ActivityID](#) ([out, retval] BSTR \*pVal)  
 HRESULT [EndTime](#) ([out, retval] long \*pVal)  
 HRESULT [RActivityId](#) ([out, retval] BSTR \*pVal)  
 HRESULT [PerformerId](#) ([out, retval] BSTR \*pVal)  
 HRESULT [Performers](#) ([out, retval] [IComPerformerList](#) \*\*pVal)  
 HRESULT [Performers](#) ([in] [IComPerformerList](#) \*pVal)  
 HRESULT [Name](#) ([out, retval] BSTR \*pVal)  
 HRESULT [GetDataFieldByName](#) ([in] BSTR bstrDataFieldName,[out, retval] [IComDataField](#) \*\*pIDataField)  
 HRESULT [GetModelActivity](#) ([out, retval] [IComModelActivity](#) \*\*pIActivity)  
 HRESULT [State](#) ([out, retval] ActivityStateEnum \*pVal)  
 HRESULT [OwnerId](#) ([out, retval] BSTR \*pVal)  
 HRESULT [IsMultiInstance](#) ([out,retval] VARIANT\_BOOL \*Result)  
 HRESULT [MultiInstances](#) ([out,retval] IComActivityList \*\*pVal)  
 HRESULT [MultiActivity](#) ([out,retval] IComActivity \*\*pVal)  
 HRESULT [Terminate](#) ([in] BSTR bstrId)  
 HRESULT [RoutingList](#) ([out, retval] IComRoutingList\* \*pVal)  
 HRESULT [IsMultiActivity](#) ([out,retval] VARIANT\_BOOL \*Result)  
 HRESULT [IsAdhocActivity](#) ([out,retval] VARIANT\_BOOL \*Result)  
 HRESULT [IsAdhocInstance](#) ([out,retval] VARIANT\_BOOL \*Result)  
 HRESULT [AdhocActivity](#) ([out, retval] IComActivity \*\*pVal)  
 HRESULT [IsTerminatable](#) ([out,retval] VARIANT\_BOOL \*Result)  
 HRESULT [AddInstanceToMultiActivity](#) ([in]BSTR bstrName, [in]BSTR bstrPerformerId,[out,retval] BSTR \* bstrInstanceId)

## Dokumentation der Elementfunktionen

HRESULT ActivityID ([out, retval] BSTR \* *pVal*)

**Beschreibung:** Diese Methode liefert die Id aus dem Workflow-Modell der Aktivität.

**Rückgabewerte:**

*BSTR*: Id aus dem Workflow-Modell

**Beispiel:** Die Id der aktuellen Aktivität wird ermittelt.

```
Id = thisprocess.CurrentActivity.ActivityID
```

HRESULT EndTime ([out, retval] long \* *pVal*)

**Beschreibung:** Diese Methode gibt an, wann die Aktivität beendet wurde.

**Rückgabewerte:**

*long*: Endzeitpunkt der Aktivität

**Beispiel:** Der Endzeitpunkt einer beliebigen Aktivität wird ermittelt.

```
Dim MyActivityID 'ActivityID aus Workflowmodell

finished =
thisprocess.GetCurrentActivityById(MyActivityID).EndTime
```

HRESULT RActivityId ([out, retval] BSTR \* *pVal*)

**Beschreibung:** Diese Methode liefert die InstanzId der Aktivität im laufenden Prozess.

**Rückgabewerte:**

*BSTR*: InstanzId

**Beispiel:** Die InstanzId der aktuellen Aktivität wird ermittelt.

```
InstanceId = thisprocess.CurrentActivity.RActivityId
```

HRESULT PerformerId ([out, retval] BSTR \* *pVal*)

**Beschreibung:** Diese Methode liefert die Id des Bearbeiters einer Aktivität.

**Rückgabewerte:**

*BSTR*: BearbeiterId

**Beispiel:** Die Bearbeiterid der aktuellen Aktivität wird ermittelt.

**Hinweis:** Dieses Script funktioniert nur im Event EndActivity.

```
PerformerId = thisprocess.CurrentActivity.PerformerId
```

HRESULT Performers ([out, retval] [IComPerformerList](#) \*\* *pVal*)

**Beschreibung:** Diese Methode liefert eine Liste aller möglichen beteiligten Rollen/Benutzer der Aktivität. Diese werden dem Modell und auch der Datenbank entnommen.

**Rückgabewerte:**

*IComPerformerList*: Liste der Beteiligten

**Beispiel:** Die Teilnehmer der aktuellen Aktivität werden ermittelt und die Liste wird nach einem bestimmten Sachbearbeiter (MyAdvisorId) durchsucht.

**Hinweis:** Dieses Script funktioniert nur im Event StartActivity.

```
Dim MyAdvisorId

`Teilnehmerliste erstellen
Set PerformersList = thisprocess.CurrentActivity.Performers

`Liste nach Sachbearbeiter durchsuchen
For i = 0 To PerformersList.count - 1
    IF PerformersList.item(i).Id = MyAdvisorId Then
        found = True
        Exit For
    End If
Next
```

HRESULT Performers ([in] [IComPerformerList](#) \* pVal)

**Beschreibung:** Diese Methode setzt die möglichen beteiligten Rollen/Benutzer der Aktivität.

**Parameter:**

*IComPerformerList*: Liste der Beteiligten

**Beispiel:** Die Teilnehmerliste einer Aktivität wird nach einem bestimmten Sachbearbeiter (MyAdvisorId) durchsucht. Sollte der Sachbearbeiter nicht in der Teilnehmerliste enthalten sein, wird dieser der Liste hinzugefügt.

**Hinweis:** Dieses Script funktioniert nur im Event StartActivity.

```
Dim MyAdvisorId

'Liste erstellen
Set PerformersList = thisprocess.CurrentActivity.Performers

'Liste nach Sachbearbeiter durchsuchen
For i = 0 To PerformersList.count - 1
    If PerformersList.item(i).Id = MyAdvisorId Then
        found = True
        Exit For
    End If
Next

If found <> True Then
    'Perfomerobjekt erstellen
    Set OrgObject = wforganisation.GetOrgObjectById(MyAdvisorId)
    Set NewPerformer = OrgObject.Performer

    'Performerobjekt der Liste hinzufügen
    PerformersList.add NewPerformer.id, NewPerformer

    'Performerliste neu setzen
    thisprocess.CurrentActivity.Performers = PerformersList
End If
```

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

**Beschreibung:** Diese Methode liefert den Namen der Aktivität.

**Rückgabewerte:**

*BSTR*: Name der Aktivität

**Beispiel:** Der Name einer beliebigen Aktivität (MyActivityID) wird ermittelt.

```
Dim MyActivityID `ActivityID aus Workflowmodell

AName = thisprocess.GetCurrentActivityById(MyActivityID).Name
```

HRESULT GetDataFieldByName ([in] BSTR *bstrDataFieldName*, [out, retval]

[IComDataField](#) \*\* *pIDataField*)

**Beschreibung:** Diese Methode liefert eine Workflowvariable einer Aktivität. Änderungen an der zurückgelieferten Variable werden **nicht** in die Datenbank übernommen. Wird eine Workflowvariable der aktuellen Aktivität erstellt, sind dort immer die Werte so enthalten, wie sie vor dem Events waren, da Änderungen an Variablen erst am Ende des Events in der Datenbank gespeichert werden.

**Parameter:**

*BSTR*: Name der Variable

**Rückgabewerte:**

*IComDataField*: Workflowvariable

**Beispiel:** Die Workflowvariable einer beliebigen Aktivität wird ermittelt.

```
Dim MyActivityID `ActivityID aus Workflowmodell
Dim MyVariable `Name der Workflowvariable

Set Activity = thisprocess.GetCurrentActivityById(MyActivityID)
Set wfvariable = Activity.GetDataFieldByName(MyVariable)
```

HRESULT GetModelActivity ([out, retval] [IComModelActivity](#) \*\* *pIActivity*)

**Beschreibung:** Diese Methode liefert zur Prozessaktivität die zugehörige Aktivität aus dem Workflowmodell.

**Rückgabewerte:**

*IComModelActivity*: Aktivität aus dem Modell

**Beispiel:** Die Workflowaktivität zur aktuellen Aktivität wird ermittelt.

```
Set wfactivity = thisprocess.CurrentActivity.GetModelActivity
```



HRESULT State ([out, retval] ActivityStateEnum \* pVal)

**Beschreibung:** Diese Methode liefert den Status der Aktivität.

**Rückgabewerte:**

*ActivityStateEnum*: Status der Aktivität

**Hinweis:** Der Status kann folgende Werte annehmen:

- § asInitialized = 0x01,
- § asRunning = 0x02,
- § asSuspended = 0x03,
- § asActive = 0x04,
- § asTerminated = 0x05,
- § asCompleted = 0x06,
- § asSysSuspended = 0x07,
- § asClosurePeriod = 0x08

**Beispiel:** Der Status einer beliebigen Aktivität wird ermittelt.

```
Dim MyActivityID 'ActivityID aus Workflowmodell

state = thisprocess.GetCurrentActivityById(MyActivityID).State
```

HRESULT OwnerId([out, retval] BSTR \*pVal)

**Beschreibung:** Die Methode liefert die Id des Benutzers, der den Arbeitsschritt momentan personalisiert hat. Die Id ist leer, falls zu der Aktivität kein Arbeitsschritt existiert oder dieser nicht personalisiert ist.

**Rückgabewerte:**

*BSTR*: Benutzerid

HRESULT IsMultiInstance([out,retval] VARIANT\_BOOL \*Result)

**Beschreibung:** Diese Eigenschaft gibt an, ob sich die Aktivität innerhalb einer Mehrinstanzaktivität befindet.

**Rückgabewerte:**

*VARIANT\_BOOL*: TRUE, wenn die Aktivität sich innerhalb einer Mehrinstanzaktivität befindet.

HRESULT MultiInstances([out,retval] IComActivityList \*\*pVal)

**Beschreibung:** Diese Methode liefert alle Mehrfachinstanzen einer Aktivität, die nicht abgebrochenen wurden.

**Rückgabewerte:**

*IComActivityList*: Liste der Mehrfachinstanzen der Aktivität

HRESULT MultiActivity([out,retval] IComActivity \*\*pVal)

**Beschreibung:** Diese Methode liefert die Mehrfachinstanzaktivität, in der sich die aktuelle Aktivität befindet.

**Rückgabewerte:** *IComActivity*: Liste der Mehrfachinstanzen der Aktivität

HRESULT Terminate([in] BSTR bstrId)

**Beschreibung:** Die Methode bricht eine Aktivität ab. Der Aufruf ist nur für Mehrinstanzaktivitäten möglich. Handelt es sich hierbei um die letzte laufende Mehrfachinstanz, wird der Prozess in seiner Bearbeitung fortgesetzt.

**Parameter:**

*BSTR*: Id der Aktivität

HRESULT RoutingList([out, retval] IComRoutingList\* \*pVal)

**Beschreibung:** Die Methode liefert die Laufliste der Aktivität, sofern es sich um eine Ad-hoc-aktivität handelt

**Rückgabewerte:**

*IComRoutingList*: Laufliste

HRESULT IsMultiActivity([out,retval]VARIANT\_BOOL \*Result)

**Beschreibung:** Die Methode gibt an, ob es sich bei der Aktivität um eine Mehrfachinstanzaktivität handelt.

**Rückgabewerte:**

*VARIANT\_BOOL*: True, wenn es sich um eine Mehrfachinstanzaktivität handelt.

HRESULT IsAdhocActivity([out,retval]VARIANT\_BOOL \*Result)

**Beschreibung:** Die Methode gibt an, ob es sich bei der Aktivität um eine Ad-hoc-aktivität handelt.

**Rückgabewerte:**

*VARIANT\_BOOL*: True, wenn es sich um eine Ad-hoc-aktivität handelt.

HRESULT IsAdhocInstance([out,retval]VARIANT\_BOOL \*Result)

**Beschreibung:** Die Methode gibt an, ob die Aktivität innerhalb einer Ad-hoc-aktivität läuft.

**Rückgabewerte:**

*VARIANT\_BOOL*: True, wenn es die Aktivität in einer Ad-hoc-aktivität läuft.

HRESULT AdhocActivity([out, retval] IComActivity \*\*pVal)

**Beschreibung:** Die Methode liefert die Ad-hoc-aktivität, in der diese Aktivität läuft.

**Rückgabewerte:**

*IComActivity*: Ad-hoc-aktivität

HRESULT IsTerminatable([out,retval]VARIANT\_BOOL \*Result)

**Beschreibung:** Die Methode gibt an, ob die Aktivität zu dem momentanen Zeitpunkt terminierbar ist.

**Rückgabewerte:**

*VARIANT\_BOOL*: True, wenn die Aktivität zu dem momentanen Zeitpunkt terminierbar ist.

**Beispiel:** Alle noch terminierbaren (nicht laufenden) Aktivitäten in Mehrfachinstanzaktivitäten beenden.

```
for idx = 0 to thisprocess.Activities.Count-1
  set act = thisprocess.Activities.Item(idx)
  if act.IsMultiInstance and act.IsTerminatable and (not
    act.MultiActivity is nothing) then
    thisprocess.CurrentActivity.Terminate act.RActivityId
  end if
next
```

HRESULT AddInstanceToMultiActivity([[in]BSTR bstrName, [in]BSTR  
bstrPerformerId,[out,retval] BSTR \* bstrInstancelId)

**Beschreibung:** Diese Methode fügt einer laufenden Mehrfachinstanz-Aktivität (die Aktivität, auf der die Methode aufgerufen wird) eine neue Instanz hinzu.

**Parameter:**

*BSTR*: Name der Instanz

*BSTR*: Id des Bearbeiters

*BSTR\**: Id der erstellten Instanz

## IComActivityList Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine statische Liste von Aktivitäten. Diese Schnittstelle ist von [IComCollection](#) abgeleitet.

**Elemente der Liste:** siehe [IComActivity](#)

## ICollection Schnittstellenreferenz

### Ausführliche Beschreibung

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

§ **count** : Diese Methode liefert die Anzahl der Listenelemente.

```
CountElements = MyList.count
```

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

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

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

§ **add** : Mit Hilfe dieser Methode kann ein Objekt der Liste hinzugefügt werden.

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

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

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

## IComDataField Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle dient zur Kapselung einer Workflowvariable. Hierbei kann es sich um einen atomaren Variablentyp, einen Record oder eine Liste handeln. Alle Workflowvariablen können im Script über ihren Namen angesprochen werden. Der Wert einer atomaren Variablen ergibt sich z.B.über:

```
value = Variable.Value
```

Handelt es sich bei der Variablen um ein Record, erfolgt der Zugriff folgendermaßen:

```
value = Record.MemberName.Value
```

In Listen können sich beliebig viele Objekte eines Datentyps befinden. Jedes Listenelement besitzt eine Position innerhalb der Liste und eine eindeutige Listenid. Der Zugriff erfolgt wahlweise über die Position (index) oder über die Listenid (MyListId).

#### Beispiel:

<code>n = List.Count</code>	<code>'Anzahl der Listenelemente</code>
<code>For index = 0 To n - 1</code>	
<code>Set MyItem = List.Item(index)</code>	<code>'Liest das i-te Listenelement aus</code>
<code>Next</code>	
<code>MyListId = ListElement.ListItemId</code>	<code>'Liefert die Listenid zu</code>
	<code>'einem Element aus der Liste</code>
<code>Set MyItem = List.Item(MyListId)</code>	<code>'Liefert das Listenelement</code>
	<code>'anhand der Listenid</code>

#### Zu beachten:

Die Position eines Listenelements ist nicht fest! Dies bedeutet, dass in zwei verschiedenen Skripten (Events) ein Listenelement seine Position wechseln kann. Möchte man auf ein bestimmtes Element aus einer Liste über verschiedene Skripte hinweg referenzieren, so muss dies über die Listenid geschehen.

Möchte man ein neues Element in eine Liste einfügen, so muss man dieses Element zuerst erzeugen. Dies geschieht über **CreateListItem**. Anschließend kann es dann in die Liste per **AddListItem** eingefügt werden.

```
Set NewElement = List.CreateListItem

'Hier kann das Element nun initialisiert werden

id = List.AddListItem(NewElement) 'In id steht nun die Listenid
```

Listenelemente werden über **RemoveListItem** entfernt. Hierbei kann die Position oder die Listenid des zu löschenden Elements angegeben werden.

```
List.RemoveListItem DelId 'Element mit 'DelId' wird aus der Liste
                           'entfernt
```

## Öffentliche Methoden

HRESULT [Value](#) ([out, retval] VARIANT \*pVal)  
 HRESULT [Value](#) ([in] VARIANT newVal)  
 HRESULT [Id](#) ([out, retval] BSTR \*pVal)  
 HRESULT [DefaultValue](#) ([out, retval] VARIANT \*pVal)  
 HRESULT [DataFieldType](#) ([out, retval] long \*pVal)

## Listenmethoden

Diese Methoden können nur bei Workflowvariablen in Listenform benutzt werden.

HRESULT [Item](#) ([in]VARIANT varItem,[out, retval][IComDataField](#) \*\*pVal)  
 HRESULT [Count](#) ([out, retval] long \*pVal)  
 HRESULT [AddListItem](#) ([in] [IComDataField](#) \*pIDataField,  
 [in, defaultvalue(1)]VARIANT\_BOOL bCopy,[out, retval] BSTR \*bstrId)  
 HRESULT [RemoveListItem](#) ([in] VARIANT vtItem)  
 HRESULT [CreateListItem](#) ([out, retval] [IComDataField](#) \*\*pIDataField)  
 HRESULT [ListItemId](#) ([out, retval] BSTR \*pVal)  
 HRESULT [Selection](#) ([out, retval] BOOL \*pVal)  
 HRESULT [Selection](#) ([in] BOOL newVal)  
 HRESULT [ClearList](#) ()

## Dokumentation der Elementfunktionen

HRESULT Item ([in] VARIANT *varItem*, [out, retval] [IComDataField](#) \*\* *pVal*)

**Beschreibung:** Diese Methode liefert einen Wert aus der Liste. Der Wert aus der Liste kann über den Index oder ListenId erfolgen.

**Parameter:**

*VARIANT*: ListenId/Index des Listenelements

**Rückgabewerte:**

*IComDataField*: Element aus der Liste

**Beispiel:**

```
Dim Index    'Position des Elements in der Liste
Dim ListId   'Eindeutiger Zugriffsschlüssel eines Listenelements

Set Element = MyList.item(Index)

'oder

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

**Zu beachten:**

Die Eigenschaft ist nur gültig, wenn es sich um ein Listenelement handelt.

HRESULT Count ([out, retval] long \* *pVal*)

**Beschreibung:** Diese Methode liefert die Anzahl der Elemente in der Liste.

**Rückgabewerte:**

*long*: Anzahl der Elemente

**Beispiel:**

```
CountElements = MyList.Count
```

**Zu beachten:**

Die Eigenschaft ist nur gültig, wenn es sich um ein Listenelement handelt.



HRESULT Value ([out, retval] VARIANT \* *pVal*)

**Beschreibung:** Diese Methode liefert den Wert einer Workflowvariable. Der Aufruf ist nur für atomare Variablen erlaubt.

**Rückgabewerte:**

*VARIANT*: Wert der Workflowvariablen

**Beispiel:**

```
value = MyVariable.Value
```

HRESULT Value ([in] VARIANT *newVal*)

**Beschreibung:** Diese Methode setzt den Wert einer Variablen. Der Aufruf ist nur für atomare Variablen erlaubt.

**Parameter:**

*VARIANT*: Wert der Variablen

**Beispiel:**

```
MyVariable.Value = MyValue
```

HRESULT Id ([out, retval] BSTR \* *pVal*)

**Beschreibung:** Diese Methode liefert die Id der Variablen aus dem Workflowmodell.

**Rückgabewerte:**

*BSTR*: Variablenid

**Beispiel:**

```
VId = MyVariable.Id
```

HRESULT DefaultValue ([out, retval] VARIANT \* *pVal*)

**Beschreibung:** Diese Methode liefert den Defaultwert der Variablen.

**Rückgabewerte:**

*VARIANT*: Defaultwert

**Beispiel:**

```
DefaultValue = MyVariable.DefaultValue
```

HRESULT DataFieldType ([out, retval] long \* *pVal*)

**Beschreibung:** Diese Methode liefert den Datentypen der Variablen.

**Rückgabewerte:**

*long*: Typ der Workflowvariablen

**Hinweis:** Die Variable Typ kann folgende Werte annehmen:

- § 1 = String
- § 2 = Bool
- § 3 = Long
- § 4 = Double
- § 5 = Datum
- § 128 = Record
- § 129 = Liste

**Beispiel:**

```
Type_Name = MyVariable.DataFieldType
```

HRESULT AddListItem ([in] [IComDataField](#) \* *pIDataField*, [in, defaultvalue(1)]  
VARIANT\_BOOL *bCopy*, [out, retval] BSTR \* *bstrId*)

**Beschreibung:** Diese Methode fügt einer Liste ein Element hinzu. Das Element wird nicht direkt in die Liste übernommen, sondern eine Kopie von diesem Element erstellt. Somit werden nachfolgende Änderungen am hinzugefügten Listenelement nicht in der Liste sichtbar. Wird *bcopy* (2.Parameter) auf FALSE gesetzt, werden nachträgliche Änderungen im Event am hinzugefügten Listenelement auch Liste sichtbar.

**Parameter:**

*IComDataField*: Dieses Objekt soll der Liste hinzugefügt werden.

*VARIANT\_BOOL*: True = Listenelement als Kopie einfügen, ansonsten False (Default auf TRUE gesetzt)

**Rückgabewerte:**

*BSTR*: Id des Objekts innerhalb der Liste

**Beispiel:** Listenobjekt (NewElement) wird erstellt und der Liste hinzugefügt.

```
'Listenobjekt erstellen
Set NewElement = MyList.CreateListItem 'MyList ist als List
                                         '[String] definiert

NewElement.value = "Hier kann ein String hineingeschrieben_
& werden"

'Listenobjekt der Liste hinzufügen
ListId = MyList.AddListItem(NewElement)
```

**Zu beachten:**

Die Eigenschaft ist nur gültig, wenn es sich um ein Listenelement handelt.

HRESULT RemoveListItem ([in] VARIANT *vtItem*)

**Beschreibung:** Diese Methode entfernt ein Element aus der Liste.

**Parameter:**

*VARIANT*: ListenId oder Index (Position)

**Beispiel:**

```
MyList.RemoveListItem( Index)

'oder

MyList.RemoveListItem(ListId)
```

**Zu beachten:**

Die Eigenschaft ist nur gültig, wenn es sich um ein Listenelement handelt.

HRESULT CreateListItem ([out, retval] [IComDataField](#) \*\* *pIDataField*)

**Beschreibung:** Diese Methode erzeugt ein Listenelement. Dieses Element kann der Liste dann über AddListItem hinzugefügt werden.

**Rückgabewerte:**

*IComDataField*: neues Listenelement

**Beispiel:** siehe [AddListItem](#)

**Zu beachten:**

Die Eigenschaft ist nur gültig, wenn es sich um ein Listenelement handelt.

HRESULT ListItemId ([out, retval] BSTR \* *pVal*)

**Beschreibung:** Diese Methode liefert die Id (eindeutiger Zugriffsschlüssel) eines Listenelements.

**Rückgabewerte:**

*BSTR*: ListenId

**Beispiel:**

```
ListId = MyList.item(Index).ListItemId
```

**Zu beachten:**

Die Eigenschaft ist nur gültig, wenn es sich um ein Listenelement handelt.

HRESULT Selection ([out, retval] BOOL \* *pVal*)

**Beschreibung:** Diese Methode ermittelt, ob ein Listenelement markiert wurde.

**Rückgabewerte:**

*BOOL*: Der Wert ist TRUE, wenn es sich um ein markiertes Listenelement handelt

**Beispiel:**

```
marked = MyList.item(index).Selection
```

**Zu beachten:**

Die Eigenschaft ist nur gültig, wenn es sich um ein Listenelement handelt.

HRESULT Selection ([in] BOOL *newVal*)

**Beschreibung:** Diese Methode markiert bzw. demarkiert ein Listenelement.

**Parameter:**

*BOOL*: Art der Markierung (TRUE=markiert FALSE=demakiert)

**Beispiel:**

```
`als markiert gekennzeichnet
MyList.item(index).Selection = TRUE

`Element demarkiert
MyList.item(index).Selection = FALSE
```

**Zu beachten:**

Die Eigenschaft ist nur gültig, wenn es sich um ein Listenelement handelt.

HRESULT ClearList ()

**Beschreibung:** Diese Methode entfernt alle Einträge aus einer Liste.

**Beispiel:**

```
MyList.ClearList
```

**Zu beachten:**

Die Eigenschaft ist nur gültig, wenn es sich um ein Listenelement handelt.

## IComFile Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt die Workflowakte. Zugriff auf ein Objekt dieser Schnittstelle erhält man folgendermaßen:

```
Set WFFile = thisprocess.WFFile
```

### Öffentliche Methoden

HRESULT [ProcessId](#) ([out, retval] BSTR \*pVal)  
HRESULT [ActivityId](#) ([out, retval] BSTR \*pVal)  
HRESULT [File](#) ([out, retval] [IComFileDocs](#) \*\*pVal)  
HRESULT [FileObjects](#) ([out, retval] [IComFileDocs](#) \*\*pVal)  
HRESULT [DocId](#) ([out, retval] BSTR \*pVal)  
HRESULT [ObjectId](#) ([out, retval] BSTR \*pVal)

### Dokumentation der Elementfunktionen

HRESULT ProcessId ([out, retval] BSTR \* *pVal*)

**Beschreibung:** Diese Methode liefert die Prozessid.

**Rückgabewerte:**

*BSTR*: Prozessid

**Beispiel:**

```
Process_Id = thisprocess.WFFile.ProcessId
```

HRESULT ActivityId ([out, retval] BSTR \* *pVal*)

**Beschreibung:** Diese Methode liefert die InstanzId der aktuellen Aktivität.

**Rückgabewerte:**

*BSTR*: InstanzId

**Beispiel:**

```
InstanceId = thisprocess.WFFile.ActivityId
```

HRESULT File ([out, retval] [IComFileDocs](#) \*\* pVal)

**Warnung:**

Diese Methode bitte nicht mehr verwenden, sie wurde durch [FileObjects](#) abgelöst.

HRESULT FileObjects ([out, retval] [IComFileDocs](#) \*\* pVal)

**Beschreibung:** Diese Methode liefert den Akteninhalt, also alle Dokumente die in der Akte enthalten sind.

**Rückgabewerte:**

*IComFileDocs*: Aktenobjekt

**Beispiel:**

```
Set Documents = thisprocess.WFFile.FileObjects
```

HRESULT [DocId](#) ([out, retval] BSTR \*pVal)

**Beschreibung:** Bitte nicht mehr verwenden, ist veraltet.

HRESULT [ObjectId](#) ([out, retval] BSTR \*pVal)

**Beschreibung:** Bitte nicht mehr verwenden, ist veraltet.

## IComFileDoc Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt ein Dokument aus der Workflowakte. In Serverskripten bekommt man Zugriff auf dieses Objekt nur in Listenform.

```
Set Documents = thisprocess.WFFile.FileObjects'.Item().

If Documents.count >= 1 Then

    Set Document = Documents.item(0)'wir haben ein Dokumenten Objekt

    DocId = Document.Id    ' ID ermitteln

    Type_Name = Document.ObjectType 'Typ ermitteln

    inWS = Document.IsWorkspaceObjekt 'Liegt es im Arbeitsbereich?

    'in welcher Aktivität wurde es der Akte hinzugefügt
    InstanceId = Document.CreationActivity

End If
```

### Öffentliche Methoden

HRESULT [CreationActivity](#) ([out, retval] BSTR \*pVal)  
HRESULT [Id](#) ([out, retval] BSTR \*pVal)  
HRESULT [ObjectType](#) ([out, retval] BSTR \*pVal)  
HRESULT [IsWorkspaceObjekt](#) ([out, retval] VARIANT\_BOOL \*pVal)  
HRESULT [IsWorkspaceObject](#) ([in] VARIANT\_BOOL pVal)  
HRESULT [IsWFTrayObject](#) ([out, retval] VARIANT\_BOOL \*pVal)  
HRESULT [IsWFTrayObject](#) ([in] VARIANT\_BOOL pVal)  
HRESULT [UseActiveVariant](#) ([in] VARIANT\_BOOL pVal)  
HRESULT [UseActiveVariant](#) ([out,retval] \*VARIANT\_BOOL pVal)  
HRESULT [OriginalDocId](#) ([out,retval] BSTR \* pVal)  
HRESULT [IsMoveable](#) ([out,retval] VARIANT\_BOOL \*pVal)  
HRESULT [IsMoveable](#) ([in] VARIANT\_BOOL newVal)  
HRESULT [IsDeletable](#) ([out,retval] VARIANT\_BOOL \*pVal)  
HRESULT [IsDeletable](#) ([in] VARIANT\_BOOL newVal)



## Dokumentation der Elementfunktionen

HRESULT CreationActivity ([out, retval] BSTR \* pVal)

**Beschreibung:** Diese Methode liefert die Id der Aktivität, in der dieses Dokument der Akte hinzugefügt wurde.

**Rückgabewerte:**

*BSTR*: InstanzId

**Beispiel:** siehe [oben](#)

HRESULT Id ([out, retval] BSTR \* pVal)

**Beschreibung:** Diese Methode liefert die Dokumentenid aus dem **enaio**°.

**Rückgabewerte:**

*BSTR*: DokumentenId

**Beispiel:** siehe [oben](#)

HRESULT ObjectType ([out, retval] BSTR \* pVal)

**Beschreibung:** Diese Methode liefert den Typ des Dokuments.

**Rückgabewerte:**

*BSTR*: Typ des Dokuments

**Beispiel:** siehe [oben](#)

HRESULT IsWorkspaceObjekt ([out, retval] VARIANT\_BOOL \* pVal)

**Beschreibung:** Diese Methode gibt an, ob das Objekt im Arbeitsbereich der Akte liegt.

**Rückgabewerte:**

*BOOL*: TRUE, wenn das Objekt im Arbeitsbereich liegt; ansonsten FALSE

**Beispiel:** siehe [oben](#)

HRESULT IsWorkspaceObject([in] VARIANT\_BOOL pVal)

**Beschreibung:** Die Methode verschiebt das Objekt in den Arbeitsbereich oder in den Infobereich.

**Parameter:**

*BOOL*: TRUE, wenn das Objekt in den Arbeitsbereich verschoben werden soll, FALSE, falls das Objekt in den Infobereich verschoben werden soll.

HRESULT IsWFTrayObject([out, retval] VARIANT\_BOOL \* pVal)

**Beschreibung:** Die Methode gibt an, ob das Objekt in der Workflowablage liegt.

**Rückgabewerte:**

*BOOL*: TRUE, falls das Objekt in der Workflowablage liegt.

HRESULT IsWFTrayObject([in] VARIANT\_BOOL pVal)

**Beschreibung:** Die Methode legt fest, ob das nun Objekt in der Workflowablage liegen soll.

**Parameter:** *BOOL*: TRUE, falls das Objekt nun in der Workflowablage liegt.

HRESULT UseActiveVariant([in] VARIANT\_BOOL pVal)

**Beschreibung:** Die Methode legt fest, ob für dieses Objekt die aktive Variante verwendet werden soll.

**Parameter:** *BOOL*: TRUE, falls für dieses Objekt die aktive Variante verwendet werden soll.

HRESULT UseActiveVariant([out,retval] VARIANT\_BOOL \*pVal)

**Beschreibung:** Die Methode gibt an, ob für dieses Objekt die aktive Variante verwendet wird.

**Rückgabewerte:**

*BOOL*: TRUE, falls für dieses Objekt die aktive Variante verwendet wird.

HRESULT OriginalDocId([out,retval] BSTR pVal)

**Beschreibung:** Die Methode gibt an, welches Dokument ursprünglich in die Akte gezogen wurde. Diese Id unterscheidet sich von der Objektid, wenn zwischenzeitlich eine neue Variante aktiv wurde.

**Rückgabewerte:**

*BSTR*: ID des Dokuments.

HRESULT IsMoveable([out, retval] VARIANT\_BOOL \* pVal)

**Beschreibung:** Diese Methode gibt an, ob das Dokument zwischen Infobereich und Arbeitsbereich vom Benutzer verschoben werden kann.

**Rückgabewerte:**

*VARIANT\_BOOL*: TRUE = kann verschoben werden

HRESULT IsMoveable([in] VARIANT\_BOOL newVal)

**Beschreibung:** Diese Methode setzt ein Dokument im Infobereich bzw. Arbeitsbereich fest, es kann nicht mehr vom Benutzer verschoben werden.

**Rückgabewerte:**

*VARIANT\_BOOL*: FALSE = kann nicht mehr verschoben werden TRUE = kann verschoben werden

HRESULT IsDeletable([out, retval] VARIANT\_BOOL \* pVal)

**Beschreibung:** Diese Methode gibt an, ob das Dokument vom Benutzer aus der Akte gelöscht werden kann.

**Rückgabewerte:**

*VARIANT\_BOOL*: TRUE = Objekt kann gelöscht werden

HRESULT IsDeletable([in] VARIANT\_BOOL newVal)

**Beschreibung:** Diese Methode definiert, ob das Dokument vom Benutzer aus der Akte gelöscht werden kann.

**Rückgabewerte:**

*VARIANT\_BOOL*: TRUE = Objekt kann gelöscht werden, FALSE = kann nicht gelöscht werden

## IComFileDocs Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine statische Liste aller Dokumente in der Akte. Diese Schnittstelle ist von [IComCollection](#) abgeleitet. Die Liste wird über folgenden Aufruf erstellt:

```
Set Documents = thisprocess.WFFile.FileObjects
```

**Elemente der Liste:** siehe [IComFileDoc](#)

## IComModel Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt das Workflowmodell. Zugriff aus das Objekt erhält man durch folgenden Aufruf:

```
Set wfmodel = thisprocess.WorkflowModel
```

### Öffentliche Methoden

HRESULT [Activities](#) ([out, retval] [IComModelActivityList](#) \*\*pVal)  
 HRESULT [FileResponsible](#) ([out, retval] IComOrgObject \*\*pVal)  
 HRESULT [InputVariables](#) ([out, retval] IComDataFields \*\*pVal)  
 HRESULT [StartProcess](#) ([in] IComFile \*pIFile, [in] IComDataFields \*pVal, BSTR \*pbstrId)  
 HRESULT [File](#) ([out, retval] IComFile \*\*pVal)  
 HRESULT [Name](#) ([out, retval] BSTR \*pbstrName)  
 HRESULT [Id](#) ([out, retval] BSTR \*pbstrId)

### Dokumentation der Elementfunktionen

HRESULT Activities ([out, retval] [IComModelActivityList](#) \*\* pVal)

**Beschreibung:** Diese Methode liefert eine Liste aller Aktivitäten des Modells.

**Rückgabewerte:**

*IComModelActivityList*: Liste der Aktivitäten

**Beispiel:** Es wird der Typ einer bestimmten Aktivität ermittelt.

```
Dim MyActivityName ` gesuchte Aktivität

Set ActivityList = thisprocess.WorkflowModel.Activities

For i = 0 To ActivityList.count - 1
  If ActivityList.item(i).name = MyActivityName Then
    Type_Name = ActivityList.item(i).type
    Exit For
  End If
Next
```

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

**Beschreibung:** Diese Methode liefert den Namen des Workflowmodells

**Rückgabewerte:**

*BSTR*: Name

HRESULT Id ([out, retval] BSTR \*pbstrId)

**Beschreibung:** Diese Methode liefert die Id Workflowmodells.

**Rückgabewerte:**

*BSTR:* Id des Modells

HRESULT FileResponsible ([out, retval] IComOrgObject\*\* pVal)

**Beschreibung:** Diese Methode liefert den Aktenverantwortlichen des Workflowmodells oder NOTHING, wenn kein Aktenverantwortlicher definiert ist.

**Rückgabewerte:**

*IComOrgObject:* Organisationsobjekt des Aktenverantwortlichen.

HRESULT InputVariables ([out, retval] IComDataFields\*\* pVal)

**Beschreibung:** Liefert die Eingangsparameter des Workflowmodells. Die Eingangsparameter sind in der Collection über ihren Namen ansprechbar und können direkt gefüllt werden.

**Rückgabewerte:**

*IComDataFields:* Collection der Eingabeparamter.

HRESULT File ([out, retval] IComFile\*\* pVal)

**Beschreibung:** Liefert ein leeres Aktenobjekt.

**Rückgabewerte:**

*IComFile:* Akte

HRESULT StartProcess ([in] IComFile \*pIFile, [in] IComDataFields \*pVal, BSTR \*pbstrId)

**Beschreibung:** Startet einen Prozess. Der Prozess wird mit den angegebenen Eingangsparametern und der übergebenen Akte initialisiert. Die Id des gestarteten Prozesses wird zurückgeliefert.

Die Abarbeitung des Prozesses beginnt nach der Beendigung des Events und erfolgt unabhängig vom Ausgang des Events. Dies bedeutet insbesondere, dass der Prozess auch im Fehlerfall des Events startet.

**Rückgabewerte:**

*IComFile:* Akte

*IComDataFields:* Eingangsparameter

**Rückgabewerte:**

*IComDataFields:* Collection der Eingabeparamter.

**Beispiel:** Ein Workflowmodell wird ausgewählt, Akte und Eingangsparameter werden gefüllt und ein Prozess wird gestartet.

```
'Es werden alle aktiven Workflowmodelle angefragt

set models = wfsystem.getactiveworkflowmodels

for i = 0 to models.count - 1
    set model = models.item(i)

    'Es soll ein 'Rechnungseingang' gestartet werden
```

```
if model.name = "Rechnungseingang" then

    'Die Eingangsparameter werden ermittelt und gesetzt...

    set params = model.inputvariables
    params.item("Rechnungstitel").value="Büromaterial"
    set var = params.item("wfprotocol")
    set varitem = var.createlistitem
    varitem.item("protokoll").value="Prozess durch ein Event
gestartet."
    varitem.item("benutzer").value="Server"
    var.addlistitem varitem

    'Der Akte wird die Rechnung beigelegt.

    set file = model.file
    file.AddFileObject "21","262144",true, 1

    'Prozess starten

    strProcessid =    model.startprocess(file, params)

    exit for
end if
next
```

## IComModelActivity Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Aktivität aus dem Workflowmodell. Das Objekt liefert die Teilnehmer, Name/Id und den Typ der Aktivität. Für den Zugriff auf ein ModelActivity-Objekt in Server-Events hat man folgende Möglichkeiten:

1. man erhält ein Objekt zur aktuellen Aktivität

```
Set ModelActivity = thisprocess.CurrentActivity.GetModelActivity
```

2. man erhält ein Objekt einer beliebigen Aktivität über die ID aus Workflowmodell

```
Dim MyActivityID 'ID aus Workflowmodell

Set Activity = thisprocess.GetCurrentActivityById(MyActivityID)
Set ModelActivity = Activity.GetModelActivity
```

3. man erhält ein Objekt einer beliebigen Aktivität über die InstanzId

```
Dim MyInstanceId 'InstanzId

Set Activity = thisprocess.GetRActivityById(MyInstanceId)
Set ModelActivity = Activity.GetModelActivity
```

Beim nachstehenden Aufruf erhält man eine Liste von ModelActivity-Objekten.

```
Set ActivityList = thisprocess.WorkflowModel.Activities
```

Siehe auch: [IComProcess](#), [IComActivity](#)

### Öffentliche Methoden

```
HRESULT Performers ([out, retval] IComPerformerList **pVal)
HRESULT ActivityId ([out, retval] BSTR *pVal)
HRESULT Name ([out, retval] BSTR *pVal)
HRESULT Type ([out, retval] ActivityTypeEnum *pVal)
```

### Dokumentation der Elementfunktionen

```
HRESULT Performers ([out, retval] IComPerformerList ** pVal)
```

**Beschreibung:** Diese Methode liefert eine Liste von Rollen/Benutzern, die über das Workflowmodell für die Ausführung dieser Aktivität vorgesehen sind.

**Rückgabewerte:**

*IComPerformerList*: Liste der Bearbeiter

**Beispiel:** Es wird eine Liste aller Teilnehmer erstellt, die für die Aktivität im Workflowmodell vorgesehen sind.

```
'Objekt erstellen
Set ModelActivity = thisprocess.CurrentActivity.GetModelActivity

Set Performerlist = ModelActivity.Performers
```

HRESULT ActivityId ([out, retval] BSTR \* *pVa*)

**Beschreibung:** Diese Methode liefert die Id der Aktivität.

**Rückgabewerte:**

*BSTR*: Id der Aktivität

**Beispiel:** Es wird die Id aus dem Workflowmodell zur aktuellen Aktivität ermittelt.

```
'Objekt erstellen
Set ModelActivity = thisprocess.CurrentActivity.GetModelActivity

Id = ModelActivity.ActivityId
```



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

**Beschreibung:** Diese Methode liefert den Namen der Aktivität.

**Rückgabewerte:**

*BSTR*: Name der Aktivität

**Beispiel:** Es wird eine bestimmte Aktivität gesucht und ihr Typ ermittelt.

```
Dim MyActivityName ' gesuchte Aktivität

Set ActivityList = thisprocess.WorkflowModel.Activities

For i = 0 To ActivityList.count - 1
    If ActivityList.item(i).Name = MyActivityName Then
        type = ActivityList.item(i).Type
        Exit For
    End If
Next
```

HRESULT Type ([out, retval] ActivityTypeEnum \* *pVal*)

**Beschreibung:** Diese Methode liefert den Typ der Aktivität.

**Rückgabewerte:**

*ActivityTypeEnum*: Typ der Aktivität

**Hinweis:** Der Typ kann folgende Werte annehmen:

- § atRoute = 1
- § atLoop = 2
- § atApplication = 3
- § atProcedure = 4
- § atSubFlow = 5

**Beispiel:** siehe [Name](#)

## IComModelActivityList Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine statische Liste aller Modelaktivitäten. Diese Liste ist von [IComCollection](#) abgeleitet. Die Liste wird durch folgenden Aufruf erstellt:

```
Set ActivityList = thisprocess.WorkflowModel.Activities
```

**Elemente der Liste:** siehe [IComModelActivity](#)

## IComOrgClass Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Organisationsklasse aus dem Workflowmodell. Ein OrgClass-Objekt erhält man durch folgenden Aufruf:

```
Dim MyObjectId 'Objektid aus dem Workflowmodell

Set OrgClass = wforganisation.GetOrgObjectById(MyObjectId).OrgClass
```

Siehe auch: [IComProcess](#), [IComOrgObject](#)

### Öffentliche Methoden

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

### Dokumentation der Elementfunktionen

HRESULT Id ([out, retval] BSTR \* *pVal*)  
**Beschreibung:** Diese Methode liefert die Id der Organisationsklasse.  
**Rückgabewerte:**  
*BSTR*: Id

#### Beispiel:

```
Dim MyObjectId 'Objektid aus dem Workflowmodell

Set OrgClass = wforganisation.GetOrgObjectById(MyObjectId).OrgClass

OrgClassId = OrgClass.Id
```

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

**Beschreibung:** Diese Methode liefert den Namen der Organisationsklasse.

**Rückgabewerte:**

*BSTR*: Name

**Beispiel:**

```
Dim MyObjectId 'Objektid aus dem Workflowmodell

Set OrgClass = wforganisation.GetOrgObjectById(MyObjectId).OrgClass

OrgClassName = OrgClass.Name
```

## IComOrgClassList Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine statische Liste aller Organisationsklassen. Diese Schnittstelle ist von [IComCollection](#) abgeleitet.

**Elemente der Liste:** siehe [IComOrgClass](#)

## IComOrgManagement Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt die Organisationsverwaltung. Das Objekt steht in Server-Events immer zur Verfügung und ist ansprechbar über den Namen **wfOrganisation**.

### Öffentliche Methoden

```
HRESULT GetOrgObjectById ([in] BSTR bstrId,[out, retval] IComOrgObject
**pIOrgObject)
HRESULT GetOrgObjectsByName ([in] BSTR bstrName,[out, retval]
IComOrgObjectList **pIOrgObjects)
HRESULT GetOrgObjectsByClassName ([in] BSTR bstrClassName,[out, retval]
IComOrgObjectList **pIOrgObjects)
HRESULT GetOrgUsersRoles ([out, retval] IComOrgObjectList **pList)
HRESULT ID ([out,retval] BSTR * bstrId)
```

### Dokumentation der Elementfunktionen

```
HRESULT GetOrgObjectById ([in] BSTR bstrId, [out, retval] IComOrgObject **
pIOrgObject)
```

**Beschreibung:** Diese Methode liefert ein Organisationsobjekt anhand seiner Id.

**Parameter:**

*BSTR*: Objektid

**Rückgabewerte:**

*IComOrgObject*: Objekt

**Beispiel:**

```
Dim MyObjectId 'Objektid aus dem Workflowmodell

Set Orgobject = wforganisation.GetOrgObjectById(MyObjectId)
```

HRESULT GetOrgObjectsByName ([in] BSTR *bstrName*, [out, retval] [IComOrgObjectList](#) \*\* *pIOrgObjects*)

**Beschreibung:** Diese Methode liefert Organisationsobjekte anhand des Names. In dem Namen kann '%' als Wildcard verwendet werden.

**Parameter:**

*BSTR*: Name des gesuchten Objektes

**Rückgabewerte:**

*IComOrgObjectList*: Liste von Objekten, die dem Suchnamen entsprechen

**Beispiel:**

```
Dim MyObjectName 'Name aus dem Workflowmodell

Set ListOrgobject = wforganisation.GetOrgObjectsByName(MyObjectName)
```

HRESULT GetOrgObjectsByClassName ([in] BSTR *bstrClassName*, [out, retval] [IComOrgObjectList](#) \*\* *pIObjects*)

**Beschreibung:** Diese Methode liefert Organisationsobjekte anhand des Klassennames. In dem Namen kann '%' als Wildcard verwendet werden.

**Parameter:**

*BSTR*: Klassenname der gesuchten Objekte

**Rückgabewerte:**

*IComOrgObjectList*: Liste von Objekten, die dem Suchnamen entsprechen

**Beispiel:**

```
Dim MyClassName 'Name aus dem Workflowmodell

Set ListOrgobject = wforganisation.GetOrgObjectsByClassName(MyClassName)
```

HRESULT GetOrgUsersRoles ([out, retval] [IComOrgObjectList](#) \*\* *pList*)

**Beschreibung:** Diese Methode liefert alle Personen und deren Rollen. Die Rollen, in denen sich die Person befindet, sind durch die Predecessor-Eigenschaft zugänglich.

**Rückgabewerte:**

*IComOrgObjectList*: Liste der Personen

**Beispiel:**

```
Set ListUserRoles = wforganisation.GetOrgUsersRoles
```

HRESULT ID ([out, retval] BSTR \* *bstrID*)

**Beschreibung:** Diese Methode liefert die Id der aktiven Organisation.

**Rückgabewerte:**

*BSTR*: Id

```
id= wforganisation.id
```

## IComOrgObjAttribute Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt ein Attribut eines Organisationsobjektes. In Serverskripten bekommt man Zugriff auf dieses Objekt nur in Listenform.

```
Dim MyObjectId 'Objektid aus dem Workflowmodell

Set AttributeList =
wforganisation.GetOrgObjectById(MyObjectId).Attributes

If AttributeList.count >= 1 Then
    'Organisationsobjekt hat mindestens ein Attribut
    Set Attribute = AttributeList.item(0)

    'es wird Id, Name und Wert vom ersten Attribute der Liste ermittelt
    AttId = Attribute.Id
    AttName = Attribute.Name
    AttValue = Attribute.Value
End If
```

### Öffentliche Methoden

HRESULT [Id](#) ([out, retval] BSTR \*pVal)  
HRESULT [Value](#) ([out, retval] BSTR \*pVal)  
HRESULT [Name](#) ([out, retval] BSTR \*pVal)

### Dokumentation der Elementfunktionen

HRESULT Id ([out, retval] BSTR \* *pVal*)  
**Beschreibung:** Diese Methode liefert die Id des Attributes.  
**Rückgabewerte:**  
*BSTR*: Id  
  
**Beispiel:** siehe [oben](#)



HRESULT Value ([out, retval] BSTR \* *pVal*)

**Beschreibung:** Diese Methode liefert den Wert des Attributes.

**Rückgabewerte:**

*BSTR*: Wert

**Beispiel:** siehe [oben](#)

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

**Beschreibung:** Diese Methode liefert den Namen des Attributes.

**Rückgabewerte:**

*BSTR*: Name

**Beispiel:** siehe [oben](#)

## IComOrgObjAttributeList Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine statische Liste aller Attribute eines Organisationsobjektes. Diese Schnittstelle ist von [IComCollection](#) abgeleitet. Die Liste wird über folgenden Aufruf erstellt:

```
Dim MyObjectId 'Objektid aus dem Workflowmodell  
  
Set AttributeList = _  
wforganisation.GetOrgObjectById(MyObjectId).Attributes
```

**Elemente der Liste:** siehe [IComOrgObjAttribute](#)

## IComOrgObject Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt ein Objekt aus der Organisationsstruktur des Workflowmodells. Zugriff auf ein Objekt dieser Schnittstelle erhält man folgendermaßen:

```
Dim MyObjectId    \ Id des Organisationsobjektes

Set MyOrgObject = wforganisation.GetOrgObjectById(MyObjectId)
```

Bei folgenden Aufrufen erhält man eine Liste von Organisationsobjekten:

```
Dim MyObjectName \ Name des Organisationsobjektes
Dim MyClassName  \ Name der Organisationsklasse

Set ListOrgObject = wforganisation.GetOrgObjectsByName(MyObjectName)

Set ListUsersRoles = wforganisation.GetOrgUsersRoles

Set ListOrgobject = wforganisation.GetOrgObjectsByClassName(MyClassName)
```

Siehe auch: [IComOrgManagement](#)

### Öffentliche Methoden

```
HRESULT Id ([out, retval] BSTR *pVal)
HRESULT Name ([out, retval] BSTR *pVal)
HRESULT Predecessors ([out, retval] IComOrgObjectList **pVal)
HRESULT Successors ([out, retval] IComOrgObjectList **pVal)
HRESULT Attributes ([out, retval] IComOrgObjAttributeList **pVal)
HRESULT OrgClass ([out, retval] IComOrgClass **pVal)
HRESULT GetPredecessorsByClassName ([in] BSTR bstrClassName,[in]
VARIANT_BOOL bDirectOnly,[out, retval] IComOrgObjectList **pList)
HRESULT GetSuccessorsByClassName ([in]BSTR bstrClassId,[out, retval]
IComOrgObjectList **SuccList)
HRESULT GetPredecessorsByName ([in] BSTR bstrName,[out, retval]
IComOrgObjectList **pList)
HRESULT GetSuccessorsByName ([in]BSTR bstrName,[out, retval]
IComOrgObjectList **pList)
HRESULT Absent ([out, retval] VARIANT_BOOL *pVal)
HRESULT Substitutes ([out, retval]IComOrgObjectList **pVal)
HRESULT Performer ([out, retval] IComPerformer **pVal)
HRESULT GetAttributeValueByName ([in]BSTR bstrName,[out, retval] BSTR *pVal)
```

## Dokumentation der Elementfunktionen

HRESULT Id ([out, retval] BSTR \* *pVal*)

**Beschreibung:** Diese Methode liefert die Objektid.

**Rückgabewerte:**

*BSTR*: ObjektId

**Beispiel:** Der Name eines Organisationsobjektes (MyObjectName) ist bekannt, es soll die Id (MyId) ermittelt werden.

```
Dim MyObjectName

Set ListOrgobject = wfororganisation.GetOrgObjectsByName(MyObjectName)

If ListOrgobject.count >= 1 Then
    'Liste enthält mindestens ein Objekt, vom ersten wird Id ermittelt
    MyId = ListOrgobject.item(0).Id
End If
```

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

**Beschreibung:** Diese Methode liefert den im **enaio®** - Workflow Editor definierten Objektnamen.

**Rückgabewerte:**

*BSTR*: Objektname

**Beispiel:**

```
Dim MyObjectId ' Objektid aus dem Workflowmodell

'Objekt erstellen
Set Orgobject = wfororganisation.GetOrgObjectById(MyObjectId)

ObjectName = Orgobject.Name
```

HRESULT Predecessors ([out, retval] [IComOrgObjectList](#) \*\* *pVal*)

**Beschreibung:** Diese Methode liefert eine Liste von allen direkten Vorgängerobjekten aus der Organisationsstruktur.

**Rückgabewerte:**

*IComOrgObjectList*: Liste aller Vorgängerobjekte

**Beispiel:**

```
Dim MyObjectId ' Objektid aus dem Workflowmodell

Set OrgObject = wfororganisation.GetOrgObjectById(MyObjectId)
Set PredecessorList = OrgObject.Predecessors

If PredecessorList.count >= 1 Then
    'Liste enthält mindestens ein Vorgänger, vom ersten wird der
    'Name ermittelt
    PredecessorName = PredecessorList.item(0).Name
End If
```

HRESULT Successors ([out, retval] [IComOrgObjectList](#) \*\* *pVal*)

**Beschreibung:** Diese Methode liefert eine Liste von allen direkten Nachfolgerobjekten aus der Organisationsstruktur.

**Rückgabewerte:**

*IComOrgObjectList*: Liste aller Nachfolgerobjekte

**Beispiel:** siehe [Predecessors](#)

HRESULT Attributes ([out, retval] [IComOrgObjAttributeList](#) \*\* pVal)

**Beschreibung:** Diese Methode liefert eine Liste aller Attribute eines Organisationsobjektes.

**Rückgabewerte:**

*IComOrgObjAttributeList*: Liste der Organisationsobjektattributen

**Beispiel:** Der Vorname (Firstname) eines Personenobjektes (MyObjectId) wird ermittelt.

```
Dim MyObjectId 'Objektid aus dem Workflowmodell

Set PersonAttribute = _
wforganisation.GetOrgObjectById(MyObjectId).Attributes

For i = 0 To PersonAttribute.count - 1
    If PersonAttribute.item(i).name = "Vorname" Then
        Firstname = PersonAttribute.item(i).value
        Exit For
    End If
Next
```

HRESULT OrgClass ([out, retval] [IComOrgClass](#) \*\* pVal)

**Beschreibung:** Diese Methode liefert die Organisationsklasse, der das Organisationsobjekt angehört.

**Rückgabewerte:**

*IComOrgClass*: Organisationsklasse

**Beispiel:**

```
Dim MyObjectId 'Objektid aus dem Workflowmodell

Set OrgClass = wforganisation.GetOrgObjectById(MyObjectId).OrgClass
```

HRESULT GetPredecessorsByClassName ([in] BSTR *bstrClassName*, [in]  
VARIANT\_BOOL *bDirectOnly*, [out, retval] [IComOrgObjectList](#) \*\* *pList*)

**Beschreibung:** Diese Methode liefert eine Liste aller Vorgänger des Objektes anhand des Klassennamens. Der Aufruf ist mit dem Jokerzeichen '%' möglich. Wenn die Methode mit *bDirectOnly* = True aufgerufen wird, muß der übergebene Klassenname auch in der direkten Vorgängerklasse enthalten sein.

**Parameter:**

*BSTR*: Name der Klasse

*VARIANT\_BOOL*: TRUE = es werden nur direkte Vorgänger zurückgeliefert; FALSE = es werden alle Vorgänger zurückgeliefert

**Rückgabewerte:**

*IComOrgObjectList*: Liste aller Vorgängerobjekte

**Beispiel:**

```
Dim MyObjectId 'Objektid aus dem Workflowmodell
Dim MyClassName 'Name der Organisationsklasse

Set OrgObject = wforganisation.GetOrgObjectById(MyObjectId)

Set PredecessorList = OrgObject.GetPredecessorsByClassName(MyClassName, _
True)
```

HRESULT GetSuccessorsByClassName ([in] BSTR *bstrClassId*, [out, retval]  
[IComOrgObjectList](#) \*\* *SuccList*)

**Beschreibung:** Diese Methode liefert eine Liste aller direkten Nachfolger anhand des Klassennamens. Der Aufruf ist mit dem Jokerzeichen '%' möglich.

**Parameter:**

*BSTR*: Name der Klasse

**Rückgabewerte:**

*IComOrgObjectList*: Liste aller Nachfolgerobjekte

**Beispiel:** siehe [GetPredecessorsByClassName](#)

HRESULT GetPredecessorsByName ([in] BSTR *bstrName*, [out, retval] [IComOrgObjectList](#) \*\* *pList*)

**Beschreibung:** Diese Methode liefert eine Liste aller direkten Vorgänger anhand des Objektnamens. Der Aufruf ist mit dem Jokerzeichen '%' möglich.

**Parameter:**

*BSTR*: Name des Objektes

**Rückgabewerte:**

*IComOrgObjectList*: Liste aller Vorgängerobjekte

**Beispiel:** siehe [GetPredecessorsByClassName](#)

HRESULT GetSuccessorsByName ([in] BSTR *bstrName*, [out, retval] [IComOrgObjectList](#) \*\* *pList*)

**Beschreibung:** Diese Methode liefert eine Liste aller direkten Nachfolger anhand des Objektnamens. Der Aufruf ist mit dem Jokerzeichen '%' möglich.

**Parameter:**

*BSTR*: Name des Objektes

**Rückgabewerte:**

*ComOrgObjectList*: Liste aller Nachfolgerobjekte

**Beispiel:** siehe [GetPredecessorsByClassName](#)

HRESULT Absent ([out, retval] VARIANT\_BOOL \* *pVal*)

**Beschreibung:** Diese Methode gibt an, ob dieser Benutzer abwesend gemeldet ist. Die Methode ist nur für Personenobjekte zulässig.

**Rückgabewerte:**

*VARIANT\_BOOL*: TRUE = abwesend, FALSE = anwesend

**Beispiel:**

```
Dim MyObjectId 'Objektid aus dem Workflowmodell

absent = wforganisation.GetOrgObjectById(MyObjectId).Absent
```



HRESULT Substitutes ([out, retval] [IComOrgObjectList](#) \*\* pVal)

**Beschreibung:** Diese Methode ermittelt zu einer Person oder einer Rolle die Stellvertreter.

**Rückgabewerte:**

*IComOrgObjectList*: Liste aller Stellvertreter

**Beispiel:**

```
Dim MyObjectId 'Objektid aus dem Workflowmodell

Set OrgObject = wforganisation.GetOrgObjectById(MyObjectId)

Set SubstituteList = OrgObject.Substitutes
```

HRESULT Performer ([out, retval] [IComPerformer](#) \*\* pVal)

**Beschreibung:** Diese Methode liefert ein Performerobjekt. Die Methode ist nur für Personen- und Rollenobjekte gültig.

**Rückgabewerte:**

*IComPerformer*: Performerobjekt

**Beispiel:**

```
Dim MyObjectId 'Objektid aus dem Workflowmodell

Set OrgObject = wforganisation.GetOrgObjectById(MyObjectId)

Set PerformerObj = OrgObject.Performer
```

**Zu beachten:**

Handelt es sich bei diesem Objekt um kein Personen/Rollenobjekt, so wird NULL (VB-S = NOTHING) zurückgeliefert. Es gibt keine Fehlermeldung.

**Siehe auch:** [IComActivity::Performers](#)

HRESULT GetAttributeValueByName ([in] BSTR *bstrName*, [out, retval] BSTR \* *pVal*)

**Beschreibung:** Diese Methode liefert den Wert eines Attributes anhand seines Namens.

**Parameter:**

*BSTR*: Name des Attributes

**Rückgabewerte:**

*BSTR*: Wert eines Attributes

**Beispiel:** Die E-Mail-Adresse einer beliebigen Person wird ermittelt.

```
Dim MyObjectId `Personenid aus dem Workflowmodell  
  
Set OrgObject = wforganisation.GetOrgObjectById(MyObjectId)  
  
EMail = OrgObject.GetAttributeValueByName("E-Mail")
```

## IComOrgObjectList Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Liste aller Organisationsobjekte. Diese Schnittstelle ist von [IComCollection](#) abgeleitet.

**Erstellung und Elemente der Liste:** siehe [IComOrgObject](#)

## IComPerformer Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt einen potenziellen Teilnehmer einer Workflowaktivität. Zugriff auf ein Performer-Objekt erhält man folgendermaßen:

```
Dim MyObjectId 'Objektid aus dem Workflowmodell

Set performer = wforganisation.GetOrgObjectById(MyObjectId).Performer
```

Siehe auch: [IComOrgObject::Performer](#)

Durch die folgenden Aufrufe erhält man eine Liste von Teilnehmern:

```
Set PerformerList = _
thisprocess.CurrentActivity.GetModelActivity().Performers
```

Siehe auch: [IComModelActivity::Performers](#)

```
Set PerformerList = thisprocess.CurrentActivity.Performers
```

Siehe auch: [IComActivity::Performers](#)

Mögliche Teilnehmer sind neben Personen und Rollen aus der Organisationsstruktur auch Prozessrollen und virtuelle Rollen. Zur Unterscheidung dient PerformerTypeEnum mit den Werten 1= Organisationsobjekt, 2= Prozessrolle und 3= virtuelle Rolle.

Virtuelle Rollen lassen sich über VirtualRoleTypeEnum unterscheiden in 1= Prozessersteller, 2 = Prozessbeteiligte, 3 = erster Bearbeiter und 4 = letzter Bearbeiter.

### Öffentliche Methoden

```
HRESULT Id ([out, retval] BSTR *pVal)
HRESULT OrgObject ([out, retval] IComOrgObject **pVal)
HRESULT Children ([out, retval] IComPerformerList* *pVal)
HRESULT Type ([out,retval] PerformerTypeEnum* IType)
HRESULT VirtualRoleType ([out,retval] VirtualRoleTypeEnum* IType);
HRESULT ActivityId ([out, retval] BSTR* bstrActivityId);
HRESULT Name ([out, retval] BSTR* bstrName);
```

## Dokumentation der Elementfunktionen

HRESULT Id ([out, retval] BSTR \* pVal)

**Beschreibung:** Diese Methode liefert die Id des Teilnehmers an der Workflowaktivität.

**Rückgabewerte:**

*BSTR*: Teilnehmerid

**Beispiel:** Teilnehmer der aktuellen Aktivität werden ermittelt.

```
Set PerformerList = thisprocess.CurrentActivity.Performers

If PerformerList.count >= 1 Then
    'Liste enthält mindestens ein Teilnehmerobjekt
    Set performer = PerformerList.item(0)

    PerformerId = performer.Id
End If
```

**Hinweis** funktioniert nur in der Startaktivität.

HRESULT OrgObject ([out, retval] [IComOrgObject](#) \*\* pVal)

**Beschreibung:** Diese Methode liefert das entsprechende Objekt aus der Organisationsstruktur zum Performer-Objekt.

**Rückgabewerte:**

*IComOrgObject*: Objekt aus der Organisationsstruktur.

**Beispiel:** Die Teilnehmerliste einer beliebigen Aktivität (MyActivityId) wird erstellt und vom ersten Listenelement ein Personenobjekt erstellt.

```
Dim MyActivityId ' Id der Aktivität aus dem Workflowmodel

Set Activity = thisprocess.GetCurrentActivityById(MyActivityId)
Set PerformerList = Activity.Performers

If PerformerList.count > 0 Then
    'Liste enthält mindestens ein Teilnehmerobjekt
    Set performer = PerformerList.item(0)

    Set OrgObject = performer.OrgObject
End If
```

HRESULT Children([out, retval] IComPerformerList\* \*pVal)

**Beschreibung:** Diese Methode liefert alle Objekte, die diesem Performerobjekt zugeordnet sind. Im Fall einer Organisationsrolle sind dies die zugeordneten Personen. Entsprechendes gilt für virtuelle Rollen und Prozessrollen.

**Rückgabewerte:**

*IComPerformerList*: Performerobjekte.

HRESULT Type([out,retval] PerformerTypeEnum\* IType)

**Beschreibung:** Diese Methode liefert den Typen des Performerobjekts.

**Rückgabewerte:**

*PerformerTypeEnum*: Performerobjekte.

**Hinweis:** Mögliche Werte

- § 1 = Organisationsobjekt
- § 2 = Prozessrolle
- § 3 = virtuelle Rolle

HRESULT VirtualRoleType([out,retval] VirtualRoleTypeEnum\* IType)

**Beschreibung:** Diese Methode liefert den speziellen Typen einer virtuellen Rolle (PerformerTypeEnum = 3).

**Rückgabewerte:**

*VirtualRoleTypeEnum:* Performerobjekte.

**Hinweis:** Mögliche Werte

- § 0 = undefiniert
- § 1 = Prozessersteller
- § 2 = Prozessbeteiligte
- § 3 = erster Bearbeiter
- § 4 = letzter Bearbeiter

HRESULT ActivityId([out, retval] BSTR\* bstrActivityId)

**Beschreibung:** Diese Methode liefert für virtuelle Rollen mit VirtualRoleTypeEnum 3 oder 4 die Id der zugehörigen Aktivität aus dem Workflowmodell.

**Rückgabewerte:**

*BSTR:* Id der Aktivität aus dem Workflowmodell.

HRESULT Name([out, retval] BSTR\* bstrName)

**Beschreibung:** Diese Methode liefert den Namen des Performerobjekts. Für Objekte der Organisationsstruktur ist dies der eingetragene Name. Für Prozessrollen ist es der Name, der bei CreateProcessRole angegeben wurde und für virtuelle Rollen werden die Namen vom System vergeben. Die Namen werden in der API Methode **nicht** lokalisiert.

**Rückgabewerte:**

*BSTR:* Name des Performerobjekts

## IComPerformerList Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Liste aller Teilnehmer einer Aktivität. Die Schnittstelle ist von [IComCollection](#) abgeleitet. Die Liste erhält man durch folgende Aufrufe:

```
Set PerformerList = _  
thisprocess.CurrentActivity.GetModelActivity().Performers
```

**Siehe auch:** [IComModelActivity::Performers](#)

```
Set PerformerList = thisprocess.CurrentActivity.Performers
```

**Siehe auch:** [IComActivity::Performers](#)

**Elemente der Liste:** siehe [IComPerformer](#)

**Hinweis:** funktioniert nur in der Startaktivität.

## IComProcess Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt den Workflowprozess. Mithilfe dieses Objekts kann z.B.:

- § der Prozessersteller ermittelt werden.
- § auf die bisher bearbeiteten Aktivitäten zugegriffen werden.
- § Dokumente in beliebige Prozessakten eingefügt werden.

Ein spezielles Objekt ist ‚**thisProcess**‘. Dieses Objekt steht in jedem Server-Event zu Verfügung und verweist immer auf den aktuellen Prozess.

### Öffentliche Methoden

```

HRESULT CreateTimerConfig ([out, retval] IComTimer **pITimer)
HRESULT Creator ([out, retval] BSTR *pVal)
HRESULT CurrentActivity ([out, retval] IComActivity **pVal)
HRESULT GetCurrentActivityById ([in] BSTR bstrActivityId,[out, retval] IComActivity
**pActivity)
HRESULT GetRActivityById ([in] BSTR ActivityID,[out, retval] IComActivity **pIActivity)
HRESULT GetTimerByConfigId ([in] BSTR Id,[out, retval] IComTimer **pITimer)
HRESULT GetTimerByConfigName ([in] BSTR Name,[out, retval] IComTimer **pITimer)
HRESULT GetTimerConfigByName ([in] BSTR Name,[out, retval] IComTimer **pITimer)
HRESULT InsertDocIntoFile ([in] BSTR bstrDocId,[in]BSTR bstrProcessId)
HRESULT MergeProtocol ([in] IComDataField *pIDataField)
HRESULT ProcessId ([out, retval] BSTR *pVal)
HRESULT ProcessName ([out, retval] BSTR *pVal)
HRESULT Responsibles ([in] IComOrgObjectList *newVal)
HRESULT Responsibles ([out, retval] IComOrgObjectList **pVal)
HRESULT RoutingList([in] BSTR bstrId, [out,retval] IComRoutingList **pVal)
HRESULT SetActivityPerformer ([in] BSTR bstrActivityId,[in] BSTR bstrPerformer)
HRESULT SetActivityPerformerList([in] BSTR bstrActivityId,[in] IComPerformerList*
pIComPerformerList)
HRESULT SuspendProcess()
HRESULT TimerConfigs ([out, retval] IComTimers **pVal)
HRESULT Timers ([out, retval] IComTimers **pVal)
HRESULT WFFile ([out, retval] IComFile **pVal)
HRESULT WorkflowModel ([out, retval] IComModel **pIModel)
HRESULT WorkflowName ([out, retval] BSTR *pVal)
HRESULT AddInstanceToMultiActivity([in] BSTR bstrRActivityId, [in]BSTR bstrName,
[in]BSTR bstrPerformerId,[out,retval] BSTR * bstrInstancelId)
HRESULT Priority([out, retval] long* IPriority)
HRESULT Priority([in] long IPriority)
HRESULT Roles ([out,retval] IComPerformerList ** pIProcessRoles)
HRESULT CreateProcessRole ([in] BSTR bstrName, [out, retval] IComPerformer **
pIProcessRole)
HRESULT ObjectId ([in] long IId)
HRESULT ObjectId ([out,retval] long *IId)

```





## Dokumentation der Elementfunktionen

HRESULT ProcessId ([out, retval] BSTR \* *pVal*)

**Beschreibung:** Diese Methode liefert die ProzessId.

**Rückgabewerte:**

*BSTR*: ProcessId

**Beispiel:**

```
Process_Id = thisprocess.processid
```

HRESULT GetRActivityById ([in] BSTR *ActivityID*, [out, retval] [IComActivity](#) \*\* *pIActivity*)

**Beschreibung:** Diese Methode liefert eine Aktivität zur InstanzId.

**Parameter:**

*BSTR*: InstanzId der Aktivität (relative ID)

**Rückgabewerte:**

*IComActivity*: Aktivität

**Beispiel:**

```
Dim InstanceId ` Instanzid der Aktivität  
  
Set activity = thisprocess.GetRActivityById(InstanceId)
```

HRESULT CurrentActivity ([out, retval] [IComActivity](#) \*\* *pVal*)

**Beschreibung:** Diese Methode liefert die aktuelle Aktivität, falls diese existiert.

**Rückgabewerte:**

*IComActivity*: aktuelle Aktivität

**Beispiel:**

```
Set activity = thisprocess.CurrentActivity
```

HRESULT GetCurrentActivityById ([in] BSTR *bstrActivityId*, [out, retval] [IComActivity](#) \*\*  
*pActivity*)

**Beschreibung:** Diese Methode liefert zu einer Aktivitätenid aus dem Workflowmodell die zugehörige Aktivität. Sollte es sich um eine Schleifenaktivität handeln, wird die letzte erstellte Instanz des Aktivitätsobjekts zurückgeliefert. Wenn die Aktivität noch nicht ausgeführt wurde, gibt es eine Fehlermeldung.

**Parameter:**

*BSTR*: WorkflowId der Aktivität

**Rückgabewerte:**

*IComActivity*: Aktivität

**Beispiel:**

```
Dim ActivityId ' Id der Aktivität aus dem Workflowmodel  
  
Set activity = thisprocess.GetCurrentActivityById(ActivityId)
```

HRESULT InsertDocIntoFile ([in] BSTR *bstrDocId*, [in] BSTR *bstrProcessId*)

**Beschreibung:** Diese Methode verschiebt ein Dokument aus der eigenen Akte in die Akte eines beliebigen Prozesses.

**Parameter:**

*BSTR*: Id des Dokuments, welches sich in der eignen WF-Akte befindet

*BSTR*: Prozessid

**Beispiel:**

```
Dim DocumentId 'Dokumentenid aus WF-Akte  
Dim Process_Id 'Prozessid  
  
thisprocess.InsertDocIntoFile DocumentId, Process_Id
```

HRESULT WFFile ([out, retval] [IComFile](#) \*\* pVal)

**Beschreibung:** Diese Methode liefert die Workflowakte.

**Rückgabewerte:**

*IComFile*: Workflowakte

**Beispiel:**

```
Set WF_File = thisprocess.WFFile
```

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

**Beschreibung:** Diese Methode liefert die Id des Prozesserstellers.

**Rückgabewerte:**

*BSTR*: Prozessersteller.

**Beispiel:**

```
ProcessCreator = thisprocess.Creator
```

HRESULT SetActivityPerformer ([in] BSTR *bstrActivityId*, [in] BSTR *bstrPerformer*)

**Beschreibung:** Diese Methode legt für zukünftige Aktivitäten den Bearbeiter fest. Diese Einstellung 'überstimmt' die Einstellungen aus dem Workflowmodell.

**Parameter:**

*BSTR*: Id der Aktivität

*BSTR*: Id des Bearbeiters

**Beispiel:** Nur der Prozessersteller führt die aktuelle Aktivität aus. Der Code kann im Event 'StartActivity' ausgeführt werden.

```
thisProcess.SetActivityPerformer thisProcess.CurrentActivity.Activityid, _  
thisProcess.Creator
```

HRESULT ProcessName ([out, retval] BSTR \* *pVal*)

**Beschreibung:** Diese Methode liefert den Prozessnamen. Der Prozessname setzt sich aus dem Workflownamen und dem Zähler (wie oft wurde dieser Prozess gestartet) zusammen.

**Rückgabewerte:**

*BSTR*: Prozessname

**Beispiel:**

```
Process_name = thisprocess.ProcessName
```

HRESULT WorkflowName ([out, retval] BSTR \* *pVal*)

**Beschreibung:** Diese Methode liefert den Workflownamen des aktuellen Prozesses.

**Rückgabewerte:**

*BSTR*: Workflowname

**Beispiel:**

```
Workflow_name = thisprocess.WorkflowName
```

HRESULT WorkflowModel ([out, retval] [IComModel](#) \*\* *pIModel*)

**Beschreibung:** Diese Methode liefert zum Prozess das dazugehörige Workflowmodell.

**Rückgabewerte:**

*IComModel*: Objekt des Workflowmodells

**Beispiel:**

```
Set workflow_model = thisprocess.WorkflowModel
```

HRESULT MergeProtocol ([in] [IComDataField](#) \* *pIDataField*)

**Beschreibung:** Diese Methode führt das Protokoll der aktuellen Aktivität mit den Protokollen der Vorgängeraktivitäten zusammen. Dies ist nach 'Joins' notwendig, um die Protokolleinträge aus den verschiedenen Zweigen in das aktuelle Protokoll zu überführen.

**Parameter:**

*IComDataField*: Protokollvariable

**Beispiel:** Die Workflowvariable 'MyProtokol' wurde in parallel laufenden Aktivitäten gesetzt und muss nach dem Join zusammengefügt werden.

```
thisprocess.MergeProtocol MyProtokol
```

HRESULT Responsibles ([out, retval] [IComOrgObjectList](#) \*\* *pVal*)

**Beschreibung:** Diese Methode liefert eine Liste aller Prozessverantwortlichen.

**Rückgabewerte:**

*IComOrgObjectList*: Liste der Prozessverantwortlichen

**Beispiel:** Löschen eines Prozessverantwortlichen (in diesem Fall der Prozessersteller).

```
Set MyResponsibles = thisprocess.Responsibles

For i = 0 To MyResponsibles.count - 1
    If MyResponsibles.item(i).id = thisprocess.Creator Then
        MyResponsibles.remove(i)
        thisprocess.Responsibles = MyResponsibles
    Exit For
End If
Next
```

HRESULT Responsibles ([in] [IComOrgObjectList](#) \* *newVal*)

**Beschreibung:** Diese Methode setzt die Prozessverantwortlichen.

**Parameter:**

*IComOrgObjectList*: Liste der Prozessverantwortlichen

**Beispiel:** siehe [Responsibles](#)

HRESULT CreateTimerConfig ([out, retval] [IComTimer](#) \*\* *pITimer*)

**Beschreibung:** Erstellt einen neuen Timer und füllt für Timer dieses Prozesses allgemeingültige Werte wie OrgId, WorkflowId und vergibt die TimerConfigId und TimerId.

**Rückgabewerte:**

*IComTimer*: pITimer

**Beispiel:**

```
Set myNewTimerConfig = thisprocess.CreateTimerConfig

' Setze Eigenschaften der myNewTimerConfig
' ...

' Füge neue Fristenkonfiguration in Liste aller Fristenkonfigurationen
' dieses Prozesses ein
thisprocess.TimerConfigs.Add myNewTimerConfig.TimerConfigId,
myNewTimerConfig
```

HRESULT GetTimerByConfigId ([in] BSTR *Id*, [out, retval] [IComTimer](#) \*\* *pITimer*)

**Beschreibung:** Ermittelt eine laufende Frist anhand der Id der Konfiguration.

**Parameter:**

*BSTR*: Id

**Rückgabewerte:**

*IComTimer*: pITimer

**Beispiel:**

```
Set myTimer = thisprocess.GetTimerByConfigId(ConfigId)
```

HRESULT GetTimerByConfigName ([in] BSTR *Name*, [out, retval] [IComTimer](#) \*\* *pITimer*)

**Beschreibung:** Ermittelt eine laufende Frist anhand des Namens der Konfiguration.

**Parameter:**

*BSTR*: Name

**Rückgabewerte:**

*IComTimer*: pITimer

**Beispiel:**

```
Set myTimer = thisprocess.GetTimerByConfigName("FK Mahnung")
```

HRESULT GetTimerConfigByName ([in] BSTR *Name*, [out, retval] [IComTimer](#) \*\* *pITimer*)

**Beschreibung:** Ermittelt eine Fristenkonfiguration anhand des Namens.

**Parameter:**

*BSTR*: Name

**Rückgabewerte:**

*IComTimer*: pITimer

**Beispiel:**

```
Set myTimerConfig = thisprocess.GetTimerConfigByName("FK Mahnung")
```

HRESULT TimerConfigs ([out, retval] [IComTimers](#) \*\* *pVal*)

**Beschreibung:** Liefert die konfigurierten Fristen zu dem Prozess.

**Rückgabewerte:**

*IComTimer*: pITimer

HRESULT Timers ([out, retval] [IComTimers](#) \*\* *pVal*)

**Beschreibung:** Liefert die laufenden Fristen zu dem Prozess.

**Rückgabewerte:**

*IComTimer*: pITimer

HRESULT RoutingList([in] BSTR bstrId, [out,retval] [IComRoutingList](#) \*\*pVal)

**Beschreibung:** Liefert die angegebene Laufliste. Die Id entspricht der Id der Ad-hoc-aktivität im Modell.

**Parameter:**

*BSTR*: Id der Ad-hoc-aktivität im Modell

**Rückgabewerte:**

*IComRoutingList*: pVal

**Beispiel:**

```
set Modell = thisProcess.WorkflowModel
set adhocAktivitaet = Modell.activities.item(2)
set Laufliste = thisProcess.RoutingList(adhocAktivitaet.activityid)
```

HRESULT SetActivityPerformerList([in] BSTR bstrActivityId,[in] [IComPerformerList](#) \*  
pIComPerformerList)

**Beschreibung:** Diese Methode legt für zukünftige Aktivitäten eine Menge von Bearbeitern fest. Diese Einstellung 'überstimmt' die Einstellungen aus dem Workflowmodell.

**Parameter:**

*BSTR*: Id der Aktivität

*IComPerformerList*: Teilnehmer



HRESULT SuspendProcess()

**Beschreibung:** Hält den Prozess an.

**Beispiel:**

```
thisProcess.SuspendProcess
```

HRESULT AddInstanceToMultiActivity([in] BSTR bstrRActivityId, [in]BSTR bstrName, [in]BSTR bstrPerformerId,[out,retval] BSTR \* bstrInstancelId)

**Beschreibung:** Diese Methode fügt einer laufenden Mehrfachinstanz-Aktivität eine neue Instanz hinzu.

**Parameter:**

*BSTR*: Id der Mehrfachinstanz-Aktivität im Prozess

*BSTR*: Name der Instanz

*BSTR*: Id des Bearbeiters

*BSTR\**: Id der erstellten Instanz

HRESULT Priority([out, retval] long\* IPriority)

**Beschreibung:** Diese Methode liefert die Verarbeitungspriorität des Prozesses. Dabei handelt es sich um einen Wert zwischen 0 und 99.

**Rückgabewerte:**

*long*: Priorität

HRESULT Priority([in] long IPriority)

**Beschreibung:** Diese Methode setzt die Verarbeitungspriorität des Prozesses. Dabei handelt es sich um einen Wert zwischen 0 und 99.

**Parameter:**

*long*: Priorität

HRESULT Roles ([out,retval] IComPerformerList \*\* pIProcessRoles)

**Beschreibung:** Diese Methode liefert alle Rollen des Prozesses. Dazu gehören Prozessrollen und virtuelle Rollen. Virtuelle Rollen haben den Typen `PERFORMER_VIRTUALROLE`, Prozessrollen haben den Typen `TYPE_PROCESSROLE`.

**Rückgabewerte:**

IComPerformerList: Rollen

HRESULT CreateProcessRole ([in] BSTR bstrName, [out, retval] IComPerformer \*\* pIProcessRole)

**Beschreibung:** Diese Methode erzeugt eine Prozessrolle.

**Parameter:**

*BSTR*: Name der Prozessrolle

**Rückgabewerte:**

*IComPerformer*: Objekt der Prozessrolle.

HRESULT ObjectId([out, retval] long\* lId)

**Beschreibung:** Die Methode liefert die Id des Dokuments, das von den Clients in der Vorschau angezeigt werden soll.

**Rückgabewerte:**

*long*: Objektid

HRESULT ObjectId([in] long lId)

**Beschreibung:** Die Methode setzt die Id des Dokuments, das von den Clients in der Vorschau angezeigt werden soll.

**Parameter:**

*long*: Objektid

## IComTerminateActivityInfo Schnittstellenreferenz

### Ausführliche Beschreibung

Dieses Objekt liefert Informationen über abzubrechende Aktivitäten. Das Objekt steht nur innerhalb eines TerminateActivity-Events zur Verfügung.

Das TerminateActivity-Event wird ausgelöst, wenn eine Aktivität bzw. ein Prozess abgebrochen werden. Wird ein ganzer Prozess abgebrochen, so wird dieses Event für jede laufende Aktivität ausgelöst. Andernfalls wird das Event lediglich für die betreffende Aktivität ausgeführt.

Das Objekt liefert die Informationen über alle Aktivitäten, die abgebrochen werden bzw. wurden und erlaubt außerdem die Ausführung weiterer Events zu verhindern.

### Öffentliche Methoden

```
HRESULT HandledTerminatedActivities ([out, retval] IComActivityList **pVal)
HRESULT StopEvents ([out, retval]
VARIANT_BOOL IComActivityList Schnittstellenrefe **pVal)
HRESULT StopEvents ([in] VARIANT_BOOL *pVal)
HRESULT UnHandledTerminatedActivities ([out, retval] IComActivityList **pVal)
```

### Dokumentation der Elementfunktionen

```
HRESULT IComTerminateActivityInfo::HandledTerminatedActivities ([out, retval]
IComActivityList **pVal)
```

**Beschreibung:** Liefert alle Aktivitäten, die abgebrochen werden sollen und für die das TerminateActivity-Event noch nicht aufgerufen wurde. Diese Liste wird nur dann mehr als eine Aktivität enthalten, wenn ein Prozess mit mehreren laufenden Aktivitäten abgebrochen wird. Andernfalls ist in dieser Liste nur eine Aktivität enthalten.

#### Rückgabewerte:

[IComActivityList](#):pVal

```
HRESULT IComTerminateActivityInfo::StopEvents ([out, retval] VARIANT_BOOL ** pVal)
```

**Beschreibung:** Liefert alle Aktivitäten, die abgebrochen werden sollen und für die das TerminateActivity-Event noch nicht aufgerufen wurde. Diese Liste wird nur dann mehr als eine Aktivität enthalten, wenn ein Prozess mit mehreren laufenden Aktivitäten abgebrochen wird. Andernfalls ist in dieser Liste nur eine Aktivität enthalten.

#### Rückgabewerte:

*VARIANT\_BOOL* [IComActivityList Schnittstellenrefe](#):pVal

```
HRESULT IComTerminateActivityInfo::StopEvents ([in] StopEvents ** pVal)
```

**Beschreibung:** Liefert alle Aktivitäten, die abgebrochen werden sollen und für die das TerminateActivity-Event noch nicht aufgerufen wurde. Diese Liste wird nur dann mehr als eine Aktivität enthalten, wenn ein Prozess mit mehreren laufenden Aktivitäten abgebrochen wird. Andernfalls ist in dieser Liste nur eine Aktivität enthalten.

#### Parameter:

*VARIANT\_BOOL*:pVal

HRESULT IComTerminateActivityInfo::UnHandledTerminatedActivities ([out, retval]  
[IComActivityList](#)\*\*pVal)

**Beschreibung:** Liefert alle Aktivitäten, die bereits abgebrochen wurden. Diese Liste wird nur dann eine Aktivität enthalten, wenn ein Prozess mit mehreren laufenden Aktivitäten abgebrochen wird. Andernfalls ist in dieser Liste leer.

**Rückgabewerte:**

[IComActivityList](#):pVal

## IComTimer Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Frist die für einen Prozess definiert ist. Es wird dadurch entweder eine Konfiguration einer Frist beschrieben, oder eine tatsächlich gestartete Frist. Konfigurationen sind entweder für das Modell angelegt und können nicht verändert werden, oder sie sind dynamisch für den Prozess angelegt und können somit auch für diesen Prozess verändert werden.

Es bestehen folgende Möglichkeiten des Zugriffs auf Konfigurationen von Fristen:

#### 1. Anhand des Namens der Konfiguration

```
Set myTimerConfig = thisprocess.GetTimerConfigByName( "FK Mahnung" )
```

#### 2. Anhand der Id der Konfiguration

```
Set myTimerConfig = thisprocess.TimerConfigs.Item(Id)
```

#### 3. Durch Iteration über alle Fristenkonfigurationen eines Prozesses (also anhand des Index)

```
For i = 0 To thisprocess.TimerConfigs.count - 1
    Set myTimerConfig = thisprocess.TimerConfigs.Item(i)
Next
```

Es bestehen folgende Möglichkeiten des Zugriffs auf gestartete Fristen:

#### 1. Anhand des Namens der Konfiguration

```
Set myTimer = thisprocess.GetTimerByConfigName( "FK Mahnung" )
```

#### 2. Anhand der Id der Konfiguration

```
Set myTimer = thisprocess.GetTimerByConfigId(Id)
```

#### 3. Anhand der Id der Frist

```
Set myTimer = thisprocess.Timers.Item(Id)
```

#### 4. Durch Iteration über alle Fristen eines Prozesses

```
For i = 0 To thisprocess.Timers.count - 1
    Set myTimer = thisprocess.Timers.Item(i)
Next
```

Eine neue Konfiguration für eine Frist wird durch folgenden Aufruf erzeugt:

```
' Erzeuge neue Fristenkonfiguration
Set myNewTimerConfig = thisprocess.CreateTimerConfig

' Setze Eigenschaften der myNewTimerConfig
' ...

' Füge neue Fristenkonfiguration in Liste aller Fristenkonfigurationen
' dieses Prozesses ein
thisprocess.TimerConfigs.Add myNewTimerConfig.TimerConfigId,
myNewTimerConfig
```

Eine laufende Frist wird durch folgenden Aufruf gelöscht:

```
thisprocess.Timers.Remove myTimer.TimerId
```

Eine Fristenkonfiguration wird durch folgenden Aufruf gelöscht:

```
thisprocess.TimerConfigs.Remove myTimerConfig.TimerConfigId
```

*Es ist zu beachten, dass einige Eigenschaften/Methoden nicht für alle Instanzen zulässig sind. Bei den Beschreibungen der betreffenden Eigenschaften/Methoden ist ein entsprechender Hinweis zu finden.*

## Öffentliche Methoden

HRESULT [Actions](#) ([out, retval] [IComTimerActions](#) \*\*pVal)  
 HRESULT [DestinationTime](#) ([in] long newVal)  
 HRESULT [DestinationTime](#) ([out, retval] long \*pVal)  
 HRESULT [DestinationType](#) ([in] long pVal)  
 HRESULT [DestinationType](#) ([out, retval] long \*pVal)  
 HRESULT [EndActivityId](#) ([in] BSTR newVal)  
 HRESULT [EndActivityId](#) ([out, retval] BSTR \*pVal)  
 HRESULT [GetConfigCopy](#) ([out, retval] [IComTimer](#) \*\*pVal)  
 HRESULT [OrganisationId](#) ([out, retval] BSTR \*pVal)  
 HRESULT [ProcessId](#) ([out, retval] BSTR \*pVal)  
 HRESULT [RunAtOnce](#) ([in] long pVal)  
 HRESULT [RunAtOnce](#) ([out, retval] long \*pVal)  
 HRESULT [StartActivityId](#) ([in] BSTR newVal)  
 HRESULT [StartActivityId](#) ([out, retval] BSTR \*pVal)  
 HRESULT [StartType](#) ([in] long pVal)  
 HRESULT [StartType](#) ([out, retval] long \*pVal)  
 HRESULT [State](#) ([in] long pVal)  
 HRESULT [State](#) ([out, retval] long \*pVal)  
 HRESULT [TimerConfigId](#) ([out, retval] BSTR \*pVal)  
 HRESULT [TimerConfigName](#) ([in] BSTR newVal)  
 HRESULT [TimerConfigName](#) ([out, retval] BSTR \*pVal)  
 HRESULT [TimerId](#) ([out, retval] BSTR \*pVal)  
 HRESULT [TimerType](#) ([in] long pVal)  
 HRESULT [TimerType](#) ([out, retval] long \*pVal)  
 HRESULT [TimeType](#) ([in] long pVal)  
 HRESULT [TimeType](#) ([out, retval] long \*pVal)  
 HRESULT [WorkflowId](#) ([out, retval] BSTR \*pVal)

## Dokumentation der Elementfunktionen

HRESULT IComTimer::Actions ([out, retval] [IComTimerActions](#) \*\* pVal)

**Beschreibung:** Diese Eigenschaft liefert die Aktionen für Mahnfristen. *Der Aufruf ist nicht für Sperrfristen zulässig.*

**Parameter:**

*long: pVal*

**Beispiel:**

```
Set MyActions = MyTimer.Actions
```

HRESULT IComTimer::DestinationTime ([in] long newVal)

**Beschreibung:** Diese Eigenschaft setzt entweder den absoluten Endzeitpunkt der Frist oder die Zeitdauer der Frist in Sekunden - je nach TimeType. Letzteres ist nur bei Konfigurationen möglich. Bereits gestartete Fristen liefern immer den absoluten Wert. *Der Aufruf ist nicht für Fristenkonfigurationen zulässig, die für das Modell konfiguriert wurden.*

**Parameter:**

*long: newVal*

**Beispiel:**

```
' Die Frist soll eine Stunde länger laufen  
MyTimer.DestinationTime = MyTimer.DestinationTime + 3600
```

HRESULT IComTimer::DestinationTime ([out, retval] long \* pVal)

**Beschreibung:** Diese Eigenschaft liefert entweder den absoluten Endzeitpunkt der Frist oder die Zeitdauer der Frist in Sekunden - je nach TimeType. Letzteres ist nur bei Konfigurationen möglich. Bereits gestartete Fristen liefern immer den absoluten Wert.

**Rückgabewerte:**

*long: pVal*

**Beispiel:**

```
myTimeStamp = myTimer.DestinationTime
```

HRESULT IComTimer::DestinationType ([in] long *pVal*)

**Beschreibung:** Diese Eigenschaft setzt den Zieltyp einer Frist. *Der Aufruf ist nicht für Fristenkonfigurationen zulässig, die für das Modell konfiguriert wurden.*

**Parameter:**

*long*: pVal

**Hinweis:** pVal kann folgende Werte annehmen:

- § 0 = Das Ende der Frist wird auf den Start der Ziel-Aktivität gesetzt
- § 1 = Das Ende der Frist wird auf das Ende der Ziel-Aktivität gesetzt

**Beispiel:**

```
myTimer.DestinationType = 1
```

HRESULT IComTimer::DestinationType ([out, retval] long \* *pVal*)

**Beschreibung:** Diese Eigenschaft liefert den Zieltyp einer Frist.

**Rückgabewerte:**

*long*: pVal

**Hinweis:** pVal kann folgende Werte annehmen:

- § 0 = Das Ende der Frist wird auf den Start der Ziel-Aktivität gesetzt
- § 1 = Das Ende der Frist wird auf das Ende der Ziel-Aktivität gesetzt

**Beispiel:**

```
myDestinationType = MyTimer.DestinationType
```

HRESULT IComTimer::EndActivityId ([in] BSTR *newVal*)

**Beschreibung:** Diese Eigenschaft setzt die Id der Aktivität bei der die Frist beginnt. *Der Aufruf ist nicht für Fristenkonfigurationen zulässig, die für das Modell konfiguriert wurden.*

**Parameter:**

*long*: newVal

**Beispiel:**

```
myEndActivityId = MyTimer.EndActivityId
```



HRESULT IComTimer::EndActivityId ([out, retval] BSTR \* *pVal*)

**Beschreibung:** Diese Eigenschaft liefert die Id der Aktivität bei der die Frist endet.

**Rückgabewerte:**

*long*: *pVal*

**Beispiel:**

```
myEndActivityId = MyTimer.EndActivityId
```

HRESULT IComTimer::GetConfigCopy ([out, retval] [IComTimer](#) \*\* *pVal*)

**Beschreibung:** Diese Methode liefert eine Kopie der Konfiguration einer Frist.

**Parameter:**

*long*: *pVal*

**Beispiel:**

```
Set MyConfigCopy = MyTimer.GetConfigCopy()
```

HRESULT IComTimer::OrganisationId ([out, retval] BSTR \* *pVal*)

**Beschreibung:** Diese Eigenschaft liefert die OrganisationsId.

**Rückgabewerte:**

*BSTR*: *pVal*

**Beispiel:**

```
OrgId = MyTimer.OrganisationId
```

HRESULT IComTimer::ProcessId ([out, retval] BSTR \* *pVal*)

**Beschreibung:** Diese Eigenschaft liefert die ProzessId dieser Frist.

**Rückgabewerte:**

*BSTR*: *pVal*

**Beispiel:**

```
myProcessId = MyTimer.ProcessId
```

HRESULT IComTimer::RunAtOnce ([in] long pVal)

**Beschreibung:** Diese Eigenschaft gibt an, ob die Frist sofort nach Ablauf des Skripts (bzw. bei Freigabe des übergeordneten [IComProcess](#)) gestartet werden soll - unabhängig von StartType und StartActivity. *Der Aufruf ist nur für dynamisch angelegte Fristenkonfigurationen zulässig.*

**Parameter:**

*long*: pVal

**Hinweis:** pVal kann folgende Werte annehmen:

- § 1 = die Frist wird sofort gestartet
- § 0 = die Frist wird nicht sofort gestartet

**Beispiel:**

```
MyTimerConfig.RunAtOnce = 1
```

HRESULT IComTimer::RunAtOnce ([out, retval] long \* pVal)

**Beschreibung:** Diese Eigenschaft gibt an, ob die Frist sofort nach Ablauf des Skripts (bzw. bei Freigabe des übergeordneten [IComProcess](#)) gestartet werden soll - unabhängig von StartType und StartActivity. *Der Aufruf ist nur für dynamisch angelegte Fristenkonfigurationen zulässig.*

**Parameter:**

*long*: pVal

**Hinweis:** pVal kann folgende Werte annehmen:

- § 1 = die Frist wird sofort gestartet
- § 0 = die Frist wird nicht sofort gestartet

**Beispiel:**

```
myRunAtOnce = MyTimerConfig.RunAtOnce
```

HRESULT IComTimer::StartActivityId ([in] BSTR newVal)

**Beschreibung:** Diese Eigenschaft setzt die Id der Aktivität bei der die Frist beginnt. *Der Aufruf ist nur für dynamisch angelegte Fristenkonfigurationen zulässig.*

**Parameter:**

*long*: newVal

**Beispiel:**

```
myStartActivityId = MyTimer.StartActivityId
```

HRESULT IComTimer::StartActivityId ([out, retval] BSTR \* pVal)

**Beschreibung:** Diese Eigenschaft liefert die Id der Aktivität bei der die Frist beginnt.

**Rückgabewerte:**

*long*: pVal

**Beispiel:**

```
MyTimer.StartActivityId = myStartActivityId
```

HRESULT IComTimer::StartType ([in] long pVal)

**Beschreibung:** Diese Eigenschaft setzt den Starttyp einer Frist. *Der Aufruf ist nur für dynamisch angelegte Fristenkonfigurationen zulässig.*

**Parameter:**

*long*: pVal

**Hinweis:** pVal kann folgende Werte annehmen:

§ 0 = Das Ende der Frist wird auf den Start der Anfangs-Aktivität gesetzt

§ 1 = Das Ende der Frist wird auf das Ende der Anfangs-Aktivität gesetzt

**Beispiel:**

```
MyTimer.StartType = 1
```

**Hinweis:** Diese Methode lässt sich nur in der Timerkonfiguration setzen.

HRESULT IComTimer::StartType ([out, retval] long \* pVal)

**Beschreibung:** Diese Eigenschaft liefert den Starttyp einer Frist.

**Rückgabewerte:**

*long*: pVal

**Hinweis:** pVal kann folgende Werte annehmen:

§ 0 = Das Ende der Frist wird auf den Start der Anfangs-Aktivität gesetzt

§ 1 = Das Ende der Frist wird auf das Ende der Anfangs-Aktivität gesetzt

**Beispiel:**

```
myStartType = MyTimer.StartType
```

**Hinweis:** Diese Methode lässt sich nur mit einem Objekt der Timerkonfiguration auslesen.

HRESULT IComTimer::State ([in] long *pVal*)

**Beschreibung:** Diese Eigenschaft setzt den Status einer Frist. *Der Aufruf ist nur für tatsächlich gestartete Fristen (nicht für Konfigurationen) zulässig.*

**Rückgabewerte:**

*long*: pVal

**Hinweis:** pVal kann folgende Werte annehmen:

- § 0 = Aktiv
- § 1 = "PostAction"
- § 2 = Inaktiv

**Beispiel:**

```
MyTimer.State = 2 ' auf inaktiv setzen
```

HRESULT IComTimer::State ([out, retval] long \* *pVal*)

**Beschreibung:** Diese Eigenschaft liefert den Status einer Frist. *Der Aufruf ist nur für tatsächlich gestartete Fristen (nicht für Konfigurationen) zulässig.*

**Rückgabewerte:**

*long*: pVal

**Hinweis:** pVal kann folgende Werte annehmen:

- § 0 = Aktiv
- § 1 = "PostAction"
- § 2 = Inaktiv

**Beispiel:**

```
myState = MyTimer.State
```

HRESULT IComTimer::TimerConfigId ([out, retval] BSTR \* *pVal*)

**Beschreibung:** Diese Eigenschaft liefert die Id der Fristenkonfiguration.

**Parameter:**

*long*: pVal

**Beispiel:**

```
myConfigId = MyTimer.TimerConfigId
```

HRESULT IComTimer::TimerConfigName ([in] BSTR *newVal*)

**Beschreibung:** Diese Eigenschaft setzt den Namen der Fristenkonfiguration. *Der Aufruf ist nur für dynamisch angelegte Fristenkonfigurationen zulässig.*

**Parameter:**

*long*: pVal

**Beispiel:**

```
MyTimer.TimerConfigName = "Meine Mahnfrist"
```

HRESULT IComTimer::TimerConfigName ([out, retval] BSTR \* *pVal*)

**Beschreibung:** Diese Eigenschaft liefert den Namen der Fristenkonfiguration.

**Parameter:**

*long*: pVal

**Beispiel:**

```
myConfigName = MyTimer.TimerConfigName
```

HRESULT IComTimer::TimerId ([out, retval] BSTR \* *pVal*)

**Beschreibung:** Diese Eigenschaft liefert die Id einer gestarteten Frist. *Dieser Aufruf ist nicht für Konfigurationen von Fristen zulässig.*

**Rückgabewerte:**

*BSTR*: pVal

**Beispiel:**

```
MyTimerId = MyTimer.TimerId
```

HRESULT IComTimer::TimerType ([in] long *pVal*)

**Beschreibung:** Diese Eigenschaft setzt den Typ einer Frist. *Der Aufruf ist nur für dynamisch angelegte Fristenkonfigurationen zulässig.*

**Parameter:**

*long*: pVal

**Hinweis:** pVal kann folgende Werte annehmen:

- § 0 = Sperrfrist
- § 1 = Mahnfrist

**Beispiel:**

```
MyTimerConfig.TimerType = 0 ' Sperrfrist
```

HRESULT IComTimer::TimerType ([out, retval] long \* *pVal*)

**Beschreibung:** Diese Eigenschaft liefert den Typ einer Frist.

**Rückgabewerte:**

*long*: pVal

**Hinweis:** pVal kann folgende Werte annehmen:

- § 0 = Sperrfrist
- § 1 = Mahnfrist

**Beispiel:**

```
Type_name = MyTimer.TimerType
```

HRESULT IComTimer::TimeType ([in] long *pVal*)

**Beschreibung:** Diese Eigenschaft setzt den TimeType. Dieser spiegelt wieder, wie der Zeitstempel, den man mit DestinationTime erhält, zu interpretieren ist. Entweder als relativer Wert oder als absoluter Wert. Ersteres ist nur für Fristenkonfigurationen zulässig.

**Parameter:**

*long*: pVal

**Hinweis:** pVal kann folgende Werte annehmen:

§ 0 = relativ

§ 1 = absolut

**Beispiel:**

```
MyTimer.TimeType = 1
```

HRESULT IComTimer::TimeType ([out, retval] long \* *pVal*)

**Beschreibung:** Diese Eigenschaft liefert den TimeType. Dieser spiegelt wieder, wie der Zeitstempel, den man mit DestinationTime erhält, zu interpretieren ist. Entweder als relativer Wert oder als absoluter Wert. Ersteres ist nur für Fristenkonfigurationen zulässig.

**Rückgabewerte:**

*long*: pVal

**Hinweis:** pVal kann folgende Werte annehmen:

§ 0 = relativ

§ 1 = absolut

**Beispiel:**

```
myTimeType = MyTimer.TimeType
```

HRESULT IComTimer::WorkflowId ([out, retval] BSTR \* *pVal*)

**Beschreibung:** Diese Eigenschaft liefert die Workflow-Modell-Id des Prozesses dieser Frist.

**Rückgabewerte:**

*BSTR*: pVal

**Beispiel:**

```
WFModelId = MyTimer.WorkflowId
```

## IComTimerFiredInfo Schnittstellenreferenz

### Ausführliche Beschreibung

Dieses Objekt liefert Informationen über den abgelaufenen Timer (Mahnfrist), die betroffenen Aktivitäten und mögliche Reaktionen. Es steht nur in TimerFired-Events zur Verfügung.

Das TimerFired-Event wird für überschrittene Mahnfristen ausgelöst, und zwar für jede laufende Aktivität, die von dieser Frist betroffen ist. Die Reihenfolge, in das Event für diese Aktivitäten aufgerufen wird ist unbestimmt. Allerdings gibt es die Möglichkeit zu prüfen, für welche Aktivitäten das Event bereits aufgerufen bzw. noch nicht aufgerufen wurde. Außerdem kann das weitere Aufrufen des Events unterbunden werden. Es kann weiterhin beeinflusst werden, ob die mit dem Timer konfigurierten Aktionen (EMail, Stellvertretersichtbarkeiten, etc) ausgeführt werden sollen oder nicht. Zusätzlich gibt es für Arbeitsschritte die Möglichkeit, diese als Folge des Überschreitens des Timers automatisch weiter zu leiten.

### Öffentliche Methoden

HRESULT [BlockingActivities](#) ([out, retval] IComActivityList\* pVal)  
 HRESULT [ExecuteTimerActions](#) ([out, retval] VARIANT\_BOOL\* pVal)  
 HRESULT [ExecuteTimerActions](#) ([in] VARIANT\_BOOL pVal)  
 HRESULT [ForwardActivity](#) ([out, retval] VARIANT\_BOOL\* pVal)  
 HRESULT [ForwardActivity](#) ([in] VARIANT\_BOOL pVal)  
 HRESULT [HandledBlockingActivities](#) ([out, retval] IComActivityList\* pVal)  
 HRESULT [StopEvents](#) ([out, retval] VARIANT\_BOOL\* pVal)  
 HRESULT [StopEvents](#) ([in] VARIANT\_BOOL pVal)  
 HRESULT [TimerConfigId](#) ([out,retval]BSTR \*pVal)  
 HRESULT [TimerId](#) ([out,retval]BSTR \*pVal)  
 HRESULT [UnhandledBlockingActivities](#) ([out, retval] IComActivityList\* pVal)

### Dokumentation der Elementfunktionen

HRESULT IComTimerFiredInfo::BlockingActivities ([out, retval] IComActivityList\* pVal)

**Beschreibung:** Liefert eine Liste aller laufenden Aktivitäten, die für das Auslösen des Timers verantwortlich sind.

**Rückgabewerte:**

*IComActivityList:* pVal

HRESULT IComTimerFiredInfo::ExecuteTimerActions ([out, retval] VARIANT\_BOOL\* pVal)

**Beschreibung:** Gibt an, ob konfigurierten Aktionen des Timers ausgeführt werden sollen.

**Rückgabewerte:**

*VARIANT\_BOOL:* pVal (TRUE= Aktionen werden ausgeführt)

HRESULT IComTimerFiredInfo:: ExecuteTimerActions ([in] VARIANT\_BOOL pVal)

**Beschreibung:** Setzt das Ausführen von Aktionen.

**Rückgabewerte:**

*VARIANT\_BOOL:* pVal (TRUE= Aktionen werden ausgeführt)



HRESULT IComTimerFiredInfo::ForwardActivity ([out, retval] VARIANT\_BOOL\* pVal)

**Beschreibung:** Gibt an, ob der Arbeitsschritt automatisch weitergeleitet werden soll. Als Bearbeiter der Aktivität wird die Workflowengine/Server eingetragen, der das Weiterleiten ausführt.

**Rückgabewerte:**

*VARIANT\_BOOL*: pVal (TRUE= Arbeitsschritt soll automatisch weitergeleitet werden)

HRESULT IComTimerFiredInfo::ForwardActivity ([in] VARIANT\_BOOL pVal)

**Beschreibung:** Setzt das automatische Weiterleiten eines Arbeitsschrittes.

**Rückgabewerte:**

*VARIANT\_BOOL*: pVal (TRUE= Arbeitsschritt soll automatisch weitergeleitet werden)

HRESULT IComTimerFiredInfo::HandledBlockingActivities ([out, retval] IComActivityList\* pVal)

**Beschreibung:** Liefert eine Liste aller laufenden Aktivitäten, die für das Auslösen des Timers verantwortlich sind und für die das TimerFired-Event bereits ausgelöst wurde.

**Rückgabewerte:**

*IComActivityList*: pVal

HRESULT IComTimerFiredInfo::StopEvents ([out, retval] VARIANT\_BOOL\* pVal)

**Beschreibung:** Gibt an, ob für dieses TimerFired-Event auch noch für weitere Aktivitäten (aus der UnhandledActivities-Liste) aufgerufen werden soll oder nicht

**Rückgabewerte:**

*VARIANT\_BOOL*: pVal (TRUE= Event wird **NICHT** für weitere Aktivitäten aufgerufen)

HRESULT IComTimerFiredInfo:: StopEvents ([in] VARIANT\_BOOL pVal)

**Beschreibung:** Regelt das Aufrufen des TimerFired-Events für bisher nicht behandelte Aktivitäten (Aktivitäten aus der UnhandledActivities-Liste).

**Rückgabewerte:**

*VARIANT\_BOOL*: pVal (TRUE= Event wird **NICHT** für weitere Aktivitäten aufgerufen)

HRESULT IComTimerFiredInfo::TimerConfigId ([out,retval]BSTR \*pVal)

**Beschreibung:** Diese Eigenschaft liefert die Id der Konfiguration des Timers.

**Parameter:**

*BSTR*: pVal

HRESULT IComTimerFiredInfo::TimerId ([out,retval]BSTR \*pVal)

**Beschreibung:** Diese Eigenschaft liefert die Id des Timers.

**Parameter:**

*BSTR*: pVal

HRESULT IComTimerFiredInfo::UnhandledBlockingActivities ([out, retval] IComActivityList\* pVal)

**Beschreibung:** Liefert eine Liste aller laufenden Aktivitäten, die für das Auslösen des Timers verantwortlich sind und für die das TimerFired-Event noch nicht ausgelöst wurde. Die aktuelle Aktivität befindet sich während der Ausführung des Events in dieser Liste.

**Rückgabewerte:**

*IComActivityList*: pVal

## IComTimers Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Liste von Fristen die für einen Prozess definiert sind. Die Schnittstelle ist von [IComCollection](#) abgeleitet.

Eine Liste aller konfigurierten Fristen eines Prozesses erhält man durch folgenden Aufruf:

```
Set Timer_Configs = thisprocess.TimerConfigs
```

Eine Liste aller bereits gestarteten Fristen erhält man durch folgenden Aufruf:

```
Set MyTimers = thisprocess.Timers
```

Eine laufende Frist wird durch folgenden Aufruf gelöscht:

```
thisprocess.Timers.Remove myTimer.TimerId
```

Eine Fristenkonfiguration wird durch folgenden Aufruf gelöscht:

```
thisprocess.TimerConfigs.Remove myTimerConfig.TimerConfigId
```

**Elemente der Liste:** siehe [IComTimer](#)

## IComTimerActions Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt die Aktionen die beim Ablauf einer Mahnfrist ausgeführt werden. Man erhält sie durch folgenden Aufruf

```
Set MyActions = MyReminderTimer.Actions
```

Siehe auch: [IComTimer](#)

### Öffentliche Methoden

```
HRESULT CreateEmail ([out, retval] IComEmail **pIEmail)  
HRESULT Depersonalize ([in] long pVal)  
HRESULT Depersonalize ([out, retval] long *pVal)  
HRESULT Emails ([out, retval] IComEmails **pVal)  
HRESULT WorkItemForSubstitutes ([in] long pVal)  
HRESULT WorkItemForSubstitutes ([out, retval] long *pVal)  
HRESULT WorkItemPerformers ([out, retval] IComPerformerList **pVal)
```

### Dokumentation der Elementfunktionen

```
HRESULT IComTimerActions::CreateEmail ([out, retval] IComEmail ** pIEmail)
```

**Beschreibung:** Diese Methode erzeugt eine neue Email.

**Rückgabewerte:**

*long*: pIEmail

**Beispiel:**

```
Set NewEmail = MyReminderTimer.Actions.CreateEmail  
  
' Setze Eigenschaften der NewEmail  
' ...  
  
' Füge neue NewEmail in Liste aller zu versendenden Emails ein  
MyReminderTimer.Actions.Emails.Add NewEmail
```

HRESULT IComTimerActions::Depersonalize ([in] long *pVal*)

**Beschreibung:** Diese Eigenschaft gibt an, ob Arbeitsschritte bei Ablauf der Frist depersonalisiert werden.

**Parameter:**

*long*: pVal

**Hinweis:** pVal kann folgende Werte annehmen:

- § 0 = es wird nicht depersonalisiert
- § 1 = es wird depersonalisiert

**Beispiel:**

```
MyReminderTimer.Actions.Depersonalize = 1
```

HRESULT IComTimerActions::Depersonalize ([out, retval] long \* *pVal*)

**Beschreibung:** Diese Eigenschaft gibt an, ob Arbeitsschritte bei Ablauf der Frist depersonalisiert werden.

**Rückgabewerte:**

*long*: pVal

**Hinweis:** pVal kann folgende Werte annehmen:

- § 0 = es wird nicht depersonalisiert
- § 1 = es wird depersonalisiert

**Beispiel:**

```
myDepersonalize = MyReminderTimer.Actions.Depersonalize
```

HRESULT IComTimerActions::Emails ([out, retval] [IComEmails](#) \*\* *pVal*)

**Beschreibung:** Diese Eigenschaft liefert die Emails die verschickt werden, wenn die Mahnfrist abgelaufen ist.

**Rückgabewerte:**

*IComEmails*: pVal

**Beispiel:**

```
Set MyEmails = MyReminderTimer.Actions.Emails
```

HRESULT IComTimerActions::WorkItemForSubstitutes ([in] long *pVal*)

**Beschreibung:** Diese Eigenschaft gibt an, ob Arbeitsschritte bei Ablauf der Frist auch Stellvertretern zugewiesen werden sollen.

**Parameter:**

*long*: pVal

**Hinweis:** pVal kann folgende Werte annehmen:

- § 0 = Stellvertreter werden nicht zugewiesen
- § 1 = Stellvertreter werden zugewiesen

**Beispiel:**

```
MyReminderTimer.Actions.WorkItemForSubstitutes = 0
```

HRESULT IComTimerActions::WorkItemForSubstitutes ([out, retval] long \* *pVal*)

**Beschreibung:** Diese Eigenschaft gibt an, ob Arbeitsschritte bei Ablauf der Frist auch Stellvertretern zugewiesen werden sollen.

**Rückgabewerte:**

*long*: pVal

**Hinweis:** pVal kann folgende Werte annehmen:

- § 0 = Stellvertreter werden nicht zugewiesen
- § 1 = Stellvertreter werden zugewiesen

**Beispiel:**

```
myWorkItemForSubstitutes =  
MyReminderTimer.Actions.WorkItemForSubstitutes
```

HRESULT IComTimerActions::WorkItemPerformers ([out, retval] [IComPerformerList](#) \*\*  
*pVal*)

**Beschreibung:** Diese Eigenschaft liefert die Performer die zugewiesen werden, wenn die Mahnfrist abgelaufen ist.

**Rückgabewerte:**

*IComPerformerList*: pVal

**Beispiel:**

```
Set MyPerformers = MyReminderTimer.Actions.WorkItemPerformers
```

## IComEmail Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Email die im Zuge der Aktionen die beim Ablauf einer Mahnfrist ausgeführt werden, verschickt wird. Man erhält eine bestimmte EMail anhand des Index in der EMailiste durch folgenden Aufruf:

```
Set Emails = MyReminderTimer.Actions.Emails.Item(0)
```

Siehe auch: [IComEmail](#)

### Öffentliche Methoden

HRESULT [CreateRecipient](#) ([out, retval] [IComEmailRecipient](#) \*\*pIEmailRecipient)  
HRESULT [EMailForPerformers](#) ([in] long pVal)  
HRESULT [EMailForPerformers](#) ([out, retval] long \*pVal)  
HRESULT [EMailForSubstitutes](#) ([in] long pVal)  
HRESULT [EMailForSubstitutes](#) ([out, retval] long \*pVal)  
HRESULT [Recipients](#) ([out, retval] [IComEmailRecipients](#) \*\*pVal)  
HRESULT [Subject](#) ([in] BSTR newVal)  
HRESULT [Subject](#) ([out, retval] BSTR \*pVal)  
HRESULT [Text](#) ([in] BSTR newVal)  
HRESULT [Text](#) ([out, retval] BSTR \*pVal)

### Dokumentation der Elementfunktionen

HRESULT IComEmail::CreateRecipient ([out, retval] [IComEmailRecipient](#) \*\*pIEmailRecipient)

**Beschreibung:** Diese Methode erzeugt einen neuen Empfänger.

**Rückgabewerte:**

*long:* [IComEmailRecipient](#)

**Beispiel:**

```
Set NewRecipient = MyEmail.CreateRecipient

' Setze Eigenschaften des NewRecipient

' Füge NewRecipient in Liste aller Empfänger der Email ein
MyEmail.Recipients.Add NewRecipient
```

HRESULT IComEmail::EMailForPerformers ([in] long *pVal*)

**Beschreibung:** Diese Eigenschaft gibt an, ob die Mail allen Bearbeitern blockierender Aktivitäten zugestellt werden soll.

**Parameter:**

*long*: pVal

**Hinweis:** pVal kann folgende Werte annehmen:

- § 0 = Bearbeiter erhalten die EMail nicht
- § 1 = Bearbeiter erhalten die EMail

**Beispiel:**

```
MyEmail.EMailForPerformers = 1
```

HRESULT IComEmail::EMailForPerformers ([out, retval] long \* *pVal*)

**Beschreibung:** Diese Eigenschaft gibt an, ob die Mail allen Bearbeitern blockierender Aktivitäten zugestellt werden soll.

**Rückgabewerte:**

*long*: pVal

**Hinweis:** pVal kann folgende Werte annehmen:

- § 0 = Bearbeiter erhalten die EMail nicht
- § 1 = Bearbeiter erhalten die EMail

**Beispiel:**

```
myEmailForPerformers = MyEmail.EMailForPerformers
```

HRESULT IComEmail::EMailForSubstitutes ([in] long *pVal*)

**Beschreibung:** Diese Eigenschaft gibt an, ob die Mail allen Stellvertretern von Bearbeitern blockierender Aktivitäten zugestellt werden soll.

**Parameter:**

*long*: pVal

**Hinweis:** pVal kann folgende Werte annehmen:

- § 0 = Stellvertreter erhalten die EMail nicht
- § 1 = Stellvertreter erhalten die EMail

**Beispiel:**

```
MyEmail.EMailForSubstitutes = 1
```

HRESULT IComEmail::EMailForSubstitutes ([out, retval] long \* *pVal*)

**Beschreibung:** Diese Eigenschaft gibt an, ob die Mail allen Stellvertretern von Bearbeitern blockierender Aktivitäten zugestellt werden soll.

**Rückgabewerte:**

*long*: pVal

**Hinweis:** pVal kann folgende Werte annehmen:

- § 0 = Bearbeiter erhalten die EMail nicht
- § 1 = Bearbeiter erhalten die EMail

**Beispiel:**

```
myEMailForSubstitutes = MyEmail.EMailForSubstitutes
```

HRESULT IComEmail::Recipients ([out, retval] [IComEmailRecipients](#) \*\* *pVal*)

**Beschreibung:** Diese Eigenschaft liefert die Empfänger der Email.

**Rückgabewerte:**

*IComEmailRecipients*: pVal

**Beispiel:**

```
Set MyRecipients = MyEmail.Recipients
```



HRESULT IComEmail::Subject ([in] BSTR *newVal*)

**Beschreibung:** Diese Eigenschaft setzt den Betreff der Email. Es gelten die gleichen Platzhalter, wie bei der Konfiguration über den Workflow-Editor.

**Parameter:**

*BSTR*: pVal

**Beispiel:**

```
MyEmail.Subject = "Bearbeitungsfrist überschritten"
```

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

**Beschreibung:** Diese Eigenschaft liefert den Betreff der Email.

**Rückgabewerte:**

*BSTR*: pVal

**Beispiel:**

```
mySubject = MyEmail.Subject
```

HRESULT IComEmail::Text ([in] BSTR *newVal*)

**Beschreibung:** Diese Eigenschaft setzt den Textinhalt der Email. Es gelten die gleichen Platzhalter, wie bei der Konfiguration über den Workflow-Editor.

**Parameter:**

*BSTR*: pVal

**Beispiel:**

```
MyEmail.Text = "Der Arbeitsschritt X muss unverzüglich bearbeitet werden."
```

HRESULT IComEmail::Text ([out, retval] BSTR \* *pVal*)

**Beschreibung:** Diese Eigenschaft liefert den Textinhalt der Email.

**Rückgabewerte:**

*BSTR*: pVal

**Beispiel:**

```
myText = MyEmail.Text
```

## IComEmails Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Liste von Emails, die bei Ablauf einer Mahnfrist verschickt werden. Die Schnittstelle ist von [IComCollection](#) abgeleitet.

Eine Liste von Emails, die bei Ablauf einer bestimmten Mahnfrist verschickt werden, erhält man folgendermaßen:

```
Set Emails = MyReminderTimer.Actions.Emails
```

Mit dem folgenden Aufruf wird eine neue Email erzeugt.

```
Set NewEmail = MyReminderTimer.Actions.CreateEmail  
  
' Setze Eigenschaften der NewEmail  
' ...  
  
' Füge neue NewEmail in Liste aller zu versendenden Emails ein  
MyReminderTimer.Actions.Emails.Add NewEmail
```

**Elemente der Liste:** siehe [IComTimerActions](#)

## IComEmailRecipient Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt einen Emailempfänger. Einen bestimmten Empfänger einer bestimmten Email die bei Ablauf einer bestimmten Mahnfrist verschickt wird, erhält man folgendermaßen:

```
Set Recipient = MyReminderTimer.Actions.Emails.Item(0).Recipients.Item(0)
```

**Elemente der Liste:** siehe [IComEmail](#)

### Öffentliche Methoden

HRESULT [EmailAddress](#) ([out, retval] BSTR \*pVal)  
HRESULT [EmailAddress](#) ([in] BSTR newVal)  
HRESULT [UserId](#) ([out, retval] BSTR \*pVal)  
HRESULT [UserId](#) ([in] BSTR newVal)  
HRESULT [Type](#) ([out, retval] long \*pVal)

### Dokumentation der Elementfunktionen

HRESULT IComEmailRecipient::EmailAddress ([in] BSTR *newVal*)

**Beschreibung:** Diese Eigenschaft setzt die Emailadresse des Empfängers. Dabei wird der Type auf 1 gesetzt.

**Parameter:**

*BSTR*: pVal

**Beispiel:**

```
MyRecipient.EmailAddress = "harry.schmidt@domain.net"
```

HRESULT IComEmailRecipient::EmailAddress ([out, retval] BSTR \* *pVal*)

**Beschreibung:** Diese Eigenschaft liefert den Emailadresse des Empfängers falls Type = 1.

**Rückgabewerte:**

*BSTR*: pVal

**Beispiel:**

```
myEmailAddress = MyRecipient.EmailAddress
```

HRESULT IComEmailRecipient::Type ([out, retval] long \* *pVal*)

**Beschreibung:** Diese Eigenschaft liefert den Empfängertyp. Dieser spiegelt wieder, ob die Email an die gegebene Email-Adresse oder die Benutzer, die der UserId entsprechen geschickt wird.

**Rückgabewerte:**

*BSTR*: *pVal*

**Hinweis:** *pVal* kann folgende Werte annehmen:

- § 0 = Email an Benutzer, die der UserId entsprechen
- § 1 = Email an die Email-Adresse

**Beispiel:**

```
myType = MyRecipient.Type
```

HRESULT IComEmailRecipient::UserId ([in] BSTR *newVal*)

**Beschreibung:** Diese Eigenschaft setzt die Objekt-Id in der Organisationsstruktur des Empfängers. Dies kann auch die Id einer Rolle sein. Dabei wird der Type auf 0 gesetzt.

**Parameter:**

*BSTR*: *pVal*

**Beispiel:**

```
MyRecipient.UserId = ObjectId
```

HRESULT IComEmailRecipient::UserId ([out, retval] BSTR \* *pVal*)

**Beschreibung:** Diese Eigenschaft liefert den Objekt-Id in der Organisationsstruktur des Empfängers falls Type = 0. Dies kann auch die Id einer Rolle sein.

**Rückgabewerte:**

*BSTR*: *pVal*

**Beispiel:**

```
myObjectId = MyRecipient.UserId
```

## IComEmailRecipients Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Liste von Emailempfängern. Die Schnittstelle ist von [IComCollection](#) abgeleitet. Eine Liste von Emailempfängern, einer bestimmten Email die bei Ablauf einer bestimmten Mahnfrist verschickt wird, erhält man folgendermaßen:

```
Set Recipients = MyReminderTimer.Actions.Emails.Item(0).Recipients
```

**Elemente der Liste:** siehe [IComTimerActions](#)

## IComSystem Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle stellt allgemeine Systemfunktionen bereit. Es wird in Server-Events mit dem Namen **wfSystem** angesprochen.

### Öffentliche Methoden

HRESULT [SendMail](#) ([in] BSTR bstrTo,[in] BSTR bstrFrom,[in] BSTR bstrSubject,[in] BSTR bstrText,[in] BSTR bstrFiles,[out, retval] long \*IRet)  
 HRESULT [cTimestampToDateTime](#) ([in] long longTimestamp,[out, retval] BSTR \*bstrDateTime)  
 HRESULT [cDateTimeToTimestamp](#)([in] BSTR bstrDateTime, [out,retval] long \*longTimestamp)  
 HRESULT [GetRoutingListTemplate](#)([in]BSTR TemplateId, [out,retval] [IComRoutingList](#) \*\*pIRoutingList)  
  
 HRESULT [GetContainingProcessesByObjectId](#)([in] long IObjectId, [out,retval] IComProcessList\*\* pVar);  
  
 HRESULT [GetActiveWorkflowmodels](#)([in,out] IComModelList \*\*pList)

### Dokumentation der Elementfunktionen

HRESULT SendMail ([in] BSTR *bstrTo*, [in] BSTR *bstrFrom*, [in] BSTR *bstrSubject*, [in] BSTR *bstrText*, [in] BSTR *bstrFiles*, [out, retval] long \* *IRet*)

**Beschreibung:** Diese Methode verschickt eine E-Mail.

**Parameter:**

*BSTR*: Empfänger (mehrere Empfänger können über ';' getrennt werden)

*BSTR*: Sender der E-Mail

*BSTR*: Subject

*BSTR*: Inhalt der E-Mail

*BSTR*: Liste von Dateien, mit ';' getrennt

**Rückgabewerte:**

*long*: 0 = E-Mail erfolgreich versandt

**Beispiel:** Senden einer E-Mail an einen bestimmten Empfänger.

```
Dim MyObjectId 'Objektid aus dem Workflowmodell
Dim from 'Absender
Dim subject ' Betreffzeile
Dim text ' Mailinhalt
Dim file 'Anhang

'Objekt des Empfängers erstellen
Set addressee = wfOrganisation.GetOrgObjectById(MyObjectId)

address = addressee.GetAttributeValueByName("E-Mail") 'Emailadresse
'holen

sendmessage = wfSystem.SendMail(address, from, subject,text, file)
```

HRESULT cTimestampToDateTime ([in] long *longTimestamp*, [out, retval] BSTR \*  
*bstrDateTime*)

**Beschreibung:** Diese Methode konvertiert einen Zeitstempel in einen String mit dem Format (DD.MM.YYYY HH:MM:SS).

**Parameter:**

*long*: Zeitstempel

**Rückgabewerte:**

*BSTR*: Datum und Uhrzeit im Format (DD.MM.YYYY HH:MM:SS)

**Beispiel:**

```
DateTime = wfsystem.cTimestampToDateTime(MyTimestamp)
```

HRESULT cDateTimeToTimestamp([in] BSTR *bstrDateTime*, [out,retval] long  
\**longTimestamp*)

**Beschreibung:** Diese Methode konvertiert einen Datumsstring - im Format (DD.MM.YYYY HH:MM:SS) - in einen Zeitstempel (Sekunden seit 1.1.1970).

**Parameter:**

*BSTR*: Datum und Uhrzeit im Format (DD.MM.YYYY HH:MM:SS)

**Rückgabewerte:**

*long*: Zeitstempel, schlägt Konvertierung fehl wird - 1 zurückgeliefert

**Beispiel:**

```
MyTimestamp = wfsystem.cDateTimeToTimestamp("10.06.2004 16:00:00")
```

HRESULT GetRoutingListTemplate([in]BSTR TemplateId, [out,retval] [IComRoutingList](#)\*\*  
pIRoutingList)

**Beschreibung:** Diese Methode lädt eine Lauflistenvorlage.

**Parameter:**

*BSTR*: TemplateId

**Rückgabewerte:**

*IComRoutingList*: Lauflistenvorlage

HRESULT GetActiveWorkflowmodels ([out, retval] IComModelList \*\*pList)

**Beschreibung:** Liefert alle aktiven Workflowmodelle. Die Modelle sind in der Collection über ihre ID ansprechbar.

**Rückgabewerte:**

*IComModelList*: Collection der aktiven Workflowmodelle.

HRESULT GetContainingProcessesById([in] long lObjectId, [out,retval]  
IComProcessList\*\* pVar)

**Beschreibung:** Die Methode liefert alle Prozesse, in deren Akte das angegebene Objekt enthalten ist. Die Prozesse können den Status laufend, angehalten oder fehlerhaft haben, jedoch nicht beendet oder abgebrochen.

**Parameter:**

*long*: Objekt-ID

**Rückgabewerte:**

IComProcessList: Collection mit Prozessen



## IFileObject Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt ein Dokument aus dem enaio®. Zugriff auf diese Schnittstelle über:

- § [IWFFile::FileObjects\(\)](#)
- § [IWFFile::GetFileObjectByID\(\)](#)

### Öffentliche Methoden

HRESULT [ID](#) ([out, retval] long \*pVal)  
HRESULT [ObjectType](#) ([out, retval] long \*pVal)  
HRESULT [AccessRights](#) ([out, retval] DocAccessRightsEnum \*pVal)  
HRESULT [IsWorkspaceObject](#) ([out, retval] VARIANT\_BOOL \*pVal)  
HRESULT [IsWorkspaceObject](#) ([in] VARIANT\_BOOL newVal)  
HRESULT [Location](#) ([out, retval] DocLocationEnum \*pVal)  
HRESULT [IsNewObject](#) ([out, retval] VARIANT\_BOOL \*pVal)  
HRESULT [Moveable](#) ([out, retval] VARIANT\_BOOL \*pVal)  
HRESULT [Moveable](#) ([in] VARIANT\_BOOL newVal)  
HRESULT [Deleteable](#) ([out, retval] VARIANT\_BOOL \*pVal)  
HRESULT [Deleteable](#) ([in] VARIANT\_BOOL newVal)  
HRESULT [UseActiveVariant](#) ([in] VARIANT\_BOOL pVal)  
HRESULT [UseActiveVariant](#) ([out,retval] VARIANT\_BOOL \*pVal)  
HRESULT [OriginalId](#) ([out,retval] BSTR pVal)

### Dokumentation der Elementfunktionen

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

**Beschreibung:** Diese Methode liefert die eindeutige ID des Dokuments in der Akte.

**Rückgabewerte:**

*long*: Objekt-ID

**Beispiel:**

```
Dim MyDocId      'Dokumentenid aus dem DMS

Set Document = wffile.GetFileObjectByID(MyDocId)

DocID = Document.ID
```

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

**Beschreibung:** Diese Methode liefert den Typ des Dokuments in der Akte.

**Rückgabewerte:**

*long*: Objekttyp

**Beispiel:**

```
Dim MyDocId      'Dokumentenid aus dem DMS

Set Document = wffile.GetFileObjectByID(MyDocId)

DocType = Document.ObjectType
```

HRESULT AccessRights ([out, retval] DocAccessRightsEnum \* pVal)

**Beschreibung:** Diese Methode liefert die Zugriffsrechte auf das Dokument.

**Rückgabewerte:**

*DocAccessRightsEnum*: Zugriffsrecht

**Hinweis:** mögliche Zugriffsrechte

```
§ accessDenied = 0,
§ accessView = 1,
§ accessEdit = 2,
§ accessDelete = 4,
§ accessEditDataSheet = 8,
§ accessAll = 15
```

**Beispiel:**

```
Dim MyDocId      'Dokumentenid aus dem DMS

Set Document = wffile.GetFileObjectByID(MyDocId)

DocAccessRights = Document.AccessRights
```

HRESULT IsWorkspaceObject ([out, retval] VARIANT\_BOOL \* *pVal*)

**Beschreibung:** Diese Methode gibt an, ob dieses Dokument im Arbeitsbereich der Akte liegt.

**Rückgabewerte:**

*VARIANT\_BOOL*: TRUE = im Arbeitsbereich

**Beispiel:**

```
Dim MyDocId      'Dokumentenid aus dem DMS

Set Document = wffile.GetFileObjectByID(MyDocId)

inWS = Document.IsWorkspaceObject
```

HRESULT IsWorkspaceObject ([in] VARIANT\_BOOL *newVal*)

**Beschreibung:** Diese Methode verschiebt ein Dokument in den Arbeits- bzw. Infobereich.

**Parameter:**

*VARIANT\_BOOL*: TRUE = im Arbeitsbereich, FALSE = im Infobereich

**Beispiel:** Das Dokument wird in den Infobereich verschoben und dort "festgesetzt".

```
Dim MyDocId      'Dokumentenid aus dem DMS

Set Document = wffile.GetFileObjectByID(MyDocId)

Document.IsWorkspaceObject = False

Document.Moveable = False
```

HRESULT Location ([out, retval] DocLocationEnum \* *pVal*)

**Beschreibung:** Diese Methode bestimmt die Position des Dokuments im **enaio**°.

**Rückgabewerte:**

*DocLocationEnum*: Position

**Hinweis:** Mögliche Werte

§ 1 = das Dokument liegt im **enaio**°

§ 2 = das Objekt liegt lediglich in der Workflowablage

**Beispiel:**

```
Dim MyDocId      'Dokumentenid aus dem DMS

Set Document = wffile.GetFileObjectByID(MyDocId)

DocLocation = Document.Location
```

HRESULT IsNewObject ([out, retval] VARIANT\_BOOL \* *pVal*)

**Beschreibung:** Diese Methode gibt an, ob das Dokument neu angelegt wurde.

**Rückgabewerte:**

*VARIANT\_BOOL*: TRUE = neu angelegt

**Beispiel:**

```
Dim MyDocId      'Dokumentenid aus dem DMS

Set Document = wffile.GetFileObjectByID(MyDocId)

NewObj = Document.IsNewObject
```

HRESULT Moveable ([out, retval] VARIANT\_BOOL \* pVal)

**Beschreibung:** Diese Methode gibt an, ob das Dokument vom Benutzer zwischen Infobereich und Arbeitsbereich verschoben werden kann.

**Rückgabewerte:**

*VARIANT\_BOOL*: TRUE = kann verschoben werden

**Beispiel:**

```
Dim MyDocId      `Dokumentenid aus dem DMS

Set Document = wffile.GetFileObjectByID(MyDocId)

dragdrop = Document.Moveable
```

HRESULT Moveable ([in] VARIANT\_BOOL newVal)

**Beschreibung:** Diese Methode setzt ein Objekt im Infobereich bzw. Arbeitsbereich fest, es kann nicht mehr vom Benutzer verschoben werden.

**Rückgabewerte:**

*VARIANT\_BOOL*: FALSE = kann nicht mehr verschoben werden; TRUE = kann verschoben werden

**Beispiel:** siehe [IsWorkspaceObject](#)

HRESULT Deleteable ([out, retval] VARIANT\_BOOL \* pVal)

**Beschreibung:** Diese Methode gibt an, ob das Objekt vom Benutzer aus der Akte gelöscht werden kann.

**Rückgabewerte:**

*VARIANT\_BOOL*: TRUE = Objekt kann gelöscht werden

**Beispiel:**

```
Dim MyDocId      `Dokumentenid aus dem DMS

Set Document = wffile.GetFileObjectByID(MyDocId)

Erasable = Document.Deleteable
```

HRESULT Deleteable ([in] VARIANT\_BOOL newVal)

**Beschreibung:** Diese Methode definiert, ob das Objekt vom Benutzer aus der Akte gelöscht werden kann.

**Rückgabewerte:**

*VARIANT\_BOOL*: TRUE = Objekt kann gelöscht werden, FALSE = kann nicht gelöscht werden

**Beispiel:**

```
Dim MyDocId      'Dokumentenid aus dem DMS

Set Document = wffile.GetFileObjectByID(MyDocId)

Document.Deleteable = False
```

HRESULT UseActiveVariant([in] VARIANT\_BOOL pVal)

**Beschreibung:** Die Methode legt fest, ob für dieses Objekt die aktive Variante verwendet werden soll.

**Parameter:** *BOOL*: TRUE, falls für dieses Objekt die aktive Variante verwendet werden soll.

HRESULT UseActiveVariant([out,retval] VARIANT\_BOOL \*pVal)

**Beschreibung:** Die Methode gibt an, ob für dieses Objekt die aktive Variante verwendet wird.

**Rückgabewerte:**

*BOOL*: TRUE, falls für dieses Objekt die aktive Variante verwendet wird.

HRESULT OriginalId([out,retval] BSTR pVal)

**Beschreibung:** Die Methode gibt an, welches Dokument ursprünglich in die Akte gezogen wurde. Diese Id unterscheidet sich von der Objektid, wenn zwischenzeitlich eine neue Variante aktiv wurde.

**Rückgabewerte:**

*BSTR*: ID des Dokuments.

## IFileObjects Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Liste aller in der Workflowakte enthaltenen enaio®-Dokumente.

**Elemente der Liste:** siehe [IFileObject](#)

**Siehe auch:** [IWFFile](#)

## IOrganisation Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt die Organisationsstruktur und ist in Client-Events immer unter den Namen '**wfororganisation**' verfügbar. Eine Organisationsstruktur besteht aus Organisationsklassen und Organisationsobjekten, wobei jedes Organisationsobjekt zu genau einer Organisationsklasse gehört. Die Definition der Organisationsklasse bestimmt Eigenschaften und Verhalten ihrer Objekte (Attribute und mögliche Vorgänger/Nachfolgerobjekte).

#### Siehe auch:

Workflowsystemhandbuch zum Thema 'Organisationsstruktur'

### Öffentliche Methoden

```
HRESULT Id ([out, retval] BSTR *pVal)
HRESULT Name ([out, retval] BSTR *pVal)
HRESULT Active ([out, retval] VARIANT_BOOL *pVal)
HRESULT GetOrgObjectById ([in] BSTR bstrId,[out, retval] IOrgObject
**pIOrgObject)
HRESULT GetOrgObjectsByName ([in] BSTR bstrName,[out, retval] IOrgObjects
**pIOrgObjects)
HRESULT GetOrgObjectsByClassName ([in] BSTR bstrClassName,[out, retval]
IOrgObjects **pIOrgObjects)
HRESULT GetOrgUsersRoles ([out, retval] IOrgObjects **pList)
HRESULT GetUsers ([out, retval] IOrgObjects **pList)
HRESULT GetRoles ([out, retval] IOrgObjects **pList)
HRESULT GetOrgClassById ([in] BSTR bstrId, [out,retval] IOrgClass **pIOrgClass)
HRESULT GetOrgClassesByName ([in] BSTR bstrName, [out,retval] IOrgClasses
**pIOrgClasses)
```

### Dokumentation der Elementfunktionen

HRESULT Id ([out, retval] BSTR \* *pVal*)

**Beschreibung:** Diese Methode liefert die Organisationsid.

**Rückgabewerte:**

*BSTR*: Organisationsid

**Beispiel:**

```
OrgID = wfororganisation.Id
```



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

**Beschreibung:** Diese Methode liefert den Organisationsnamen.

**Rückgabewerte:**

*BSTR*: Organisationsname

**Beispiel:**

```
OrgName = wfororganisation.Name
```

HRESULT Active ([out, retval] VARIANT\_BOOL \* *pVal*)

**Beschreibung:** Diese Methode gibt an, ob die Organisation aktiv ist.

**Rückgabewerte:**

*VARIANT\_BOOL*: TRUE = Organisation aktiv

**Beispiel:**

```
varactive = wfororganisation.Active
```

HRESULT GetOrgObjectById ([in] BSTR *bstrId*, [out, retval] [IOrgObject](#) \*\* *pIOrgObject*)

**Beschreibung:** Diese Methode liefert ein Organisationsobjekt anhand seiner Id.

**Parameter:**

*BSTR*: Objektid

**Rückgabewerte:**

*IOrgObject*: Objekt

**Beispiel:**

```
Dim MyObjectId ' Id des Organisationsobjektes  
  
Set OrgObject = wfororganisation.GetOrgObjectById(MyObjectId)
```

HRESULT GetOrgObjectsByName ([in] BSTR *bstrName*, [out, retval] [IOrgObjects](#) \*\*  
*pIOrgObjects*)

**Beschreibung:** Diese Methode liefert Organisationsobjekte anhand des Names. In dem Namen kann '%' als Wildcard verwendet werden.

**Parameter:**

*BSTR*: Name des gesuchten Objektes.

**Rückgabewerte:**

*IOrgObjects*: Liste von Objekten, die dem Suchnamen entsprechen

**Beispiel:**

```
Dim MyObjectName ' Name des Organisationsobjektes  
  
Set ListOrgObject = wforganisation.GetOrgObjectsByName(MyObjectName)
```

HRESULT GetOrgObjectsByClassName ([in] BSTR *bstrClassName*, [out, retval]  
[IOrgObjects](#) \*\* *pIOrgObjects*)

**Beschreibung:** Diese Methode liefert Organisationsobjekte anhand des Klassennames. In dem Namen kann '%' als Wildcard verwendet werden.

**Parameter:**

*BSTR*: Klassenname der gesuchten Objekte

**Rückgabewerte:**

*IOrgObjects*: Liste von Objekten, die dem Suchnamen entsprechen

**Beispiel:**

```
Dim MyClassName ' Name des Organisationsobjektes  
  
Set ListOrgObject = wforganisation.GetOrgObjectsByClassName(MyClassName)
```

HRESULT GetOrgUsersRoles ([out, retval] [IOrgObjects](#) \*\* pList)

**Beschreibung:** Diese Methode liefert alle Personen und deren Rollen. Die Rollen, in denen sich die Person befindet, sind durch die Predecessor-Eigenschaft der Organisationsobjekte zugänglich.

**Rückgabewerte:**

*IOrgObjects*: Liste der Personen

**Beispiel:**

```
Set ListUserRoles = wforganisation.GetOrgUsersRoles
```

HRESULT GetUsers ([out, retval] [IOrgObjects](#) \*\* pList)

**Beschreibung:** Diese Methode liefert alle Personen-Objekte. Diese Methode entspricht dem Aufruf: 'GetOrgObjectByClassName("Person")'.

**Rückgabewerte:**

*IOrgObjects*: Liste der Benutzer

**Beispiel:**

```
Set ListUser = wforganisation.GetUsers
```

HRESULT GetRoles ([out, retval] [IOrgObjects](#) \*\* pList)

**Beschreibung:** Diese Methode liefert alle Rollen-Objekte. Diese Methode entspricht dem Aufruf: 'GetOrgObjectByClassName("Rolle")'.

**Rückgabewerte:**

*IOrgObjects*: Liste der Personen

**Beispiel:**

```
Set ListRoles = wforganisation.GetRoles
```

HRESULT GetOrgClassById ([in] BSTR bstrId, [out,retval] [IOrgClass](#) \*\*pIOrgClass)

**Beschreibung:** Diese Methode liefert eine Organisationsklasse anhand der Klassenid.

**Parameter:**

*BSTR*: Id der gesuchten Klasse

**Rückgabewerte:**

*IOrgClass*: Organisationsklasse

HRESULT GetOrgClassesByName ([in] BSTR bstrName, [out,retval] [IOrgClasses](#) \*\*pIOrgClasses)

**Beschreibung:** Diese Methode liefert Organisationsklassen anhand des Names. In dem Namen kann '%' als Wildcard verwendet werden.

**Parameter:**

*BSTR*: Name der gesuchten Klasse

**Rückgabewerte:**

*IOrgClasses*: Liste von Klassen, die dem Suchnamen entsprechen

## IOrgAttribute Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt ein Attribut eines Organisationsobjektes. In Client-Events erhält man Zugriff auf dieses Objekt nur in Listenform.

```
Dim MyObjectId ` Id des Organisationsobjektes

Set OrgObject = wforganisation.GetOrgObjectById(MyObjectId)
Set AttributeList = OrgObject.Attributes

If AttributeList.count >= 1 Then
    `Organisationsobjekt hat mindestens ein Attribut
    Set Attribute = AttributeList.item(0)
    `es wird Id, Name und Wert vom ersten Attribute der Liste ermittelt
    AttId = Attribute.Id
    AttName = Attribute.Name
    AttValue = Attribute.Value
End If
```

### Öffentliche Methoden

HRESULT [Name](#) ([out, retval] BSTR \*pVal)  
HRESULT [Value](#) ([out, retval] BSTR \*pVal)  
HRESULT [Id](#) ([out, retval] BSTR \*pVal)

### Dokumentation der Elementfunktionen

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

**Beschreibung:** Diese Methode liefert die Id des Attributes.

**Rückgabewerte:**

*BSTR*: Id

**Beispiel:** siehe [oben](#)

HRESULT Value ([out, retval] BSTR \* *pVal*)

**Beschreibung:** Diese Methode liefert den Wert des Attributes.

**Rückgabewerte:**

*BSTR*: Wert

**Beispiel:** siehe [oben](#)

HRESULT Id ([out, retval] BSTR \* *pVal*)

**Beschreibung:** Diese Methode liefert den Namen eines Attributes.

**Rückgabewerte:**

*BSTR*: Name

**Beispiel:** siehe [oben](#)

## IOrgAttributes Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine statische Liste aller Objektattribute. Diese Schnittstelle ist von [IComCollection](#) abgeleitet. Die Liste wird über folgenden Aufruf erstellt:

```
Dim MyObjectId ' Id des Organisationsobjektes  
  
Set AttributeList =  
wforganisation.GetOrgObjectById(MyObjectId).Attributes
```

**Elemente der Liste:** siehe [IOrgAttribute](#)

## IOrgClass Schnittstellenreferenz

Diese Schnittstelle beschreibt eine Organisationsklasse. Zugriff auf diese Schnittstelle erhält man folgendermaßen:

```
Dim MyClassId ` Id des Organisationsklasse  
  
Set MyOrgObject = wfororganisation.GetOrgClassById(MyClassId)
```

Bei folgenden Aufrufen erhält man eine Liste von Organisationsobjekten:

```
Dim MyClassName ` Name der Organisationsklasse  
  
Set ListOrgObject = wfororganisation.GetOrgClassesByName(MyClassName)
```

Siehe auch: [IOrganisation](#)

## Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Organisationsklasse.

## Öffentliche Methoden

HRESULT [Id](#) ([out, retval] BSTR \*pVal)  
HRESULT [Name](#) ([out, retval] BSTR \*pVal)  
HRESULT [OrganisationId](#) ([out, retval] BSTR \* pVal)



## Dokumentation der Elementfunktionen

HRESULT Id ([out, retval] BSTR \* *pVal*)

**Beschreibung:** Diese Methode liefert die Id der Organisationsklasse.

**Rückgabewerte:**

*BSTR*: Id der Klasse.

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

**Beschreibung:** Diese Methode liefert den Namen der Organisationsklasse.

**Rückgabewerte:**

*BSTR*: Name der Klasse.

HRESULT OrganisationId ([out, retval] BSTR \* *pVal*)

**Beschreibung:** Diese Methode liefert den Id der Organisation in der sich die Organisationsklasse befindet.

**Rückgabewerte:**

*BSTR*: Id der Organisation

## IOrgClasses Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Liste von Organisationsklassen. Diese Schnittstelle ist von IComCollection abgeleitet.

**Zugriff auf die Liste:** siehe [IOrganisation](#)

**Elemente der Liste:** siehe [IOrgClass](#)

## IOrgObject Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt ein Organisationsobjekt. Organisationsobjekte haben neben ihren Indexdaten (Id, Name) und Attributen auch direkte Vorgänger/Nachfolgerobjekte. Über diese wird der Aufbau der Organisationsstruktur abgebildet. Z.B. erhält man für ein Rollen-Objekt über 'Successors' sämtliche Personen-Objekte, die dieser Rolle zugeordnet sind. Ein Objekt dieser Schnittstelle wird folgendermaßen erzeugt:

```
Dim MyObjectId    ' Id des Organisationsobjektes

Set MyOrgObject = wfororganisation.GetOrgObjectById(MyObjectId)
```

Bei folgenden Aufrufen erhält man eine Liste von Organisationsobjekten:

```
Dim MyObjectName  ' Name des Organisationsobjektes
Dim MyClassName   ' Name der Organisationsklasse

Set ListOrgObject = wfororganisation.GetOrgObjectsByName(MyObjectName)

Set ListUsersRoles = wfororganisation.GetOrgUsersRoles

Set ListOrgobject = wfororganisation.GetOrgObjectsByClassName(MyClassName)

Set ListUser = wfororganisation.GetUsers

Set ListeRoles = wfororganisation.GetRoles
```

Siehe auch: [IOrganisation](#)

## Öffentliche Methoden

HRESULT [Id](#) ([out, retval] BSTR \*pVal)  
 HRESULT [Name](#) ([out, retval] BSTR \*pVal)  
 HRESULT [Predecessors](#) ([out, retval] [IOrgObjects](#) \*\*pVal)  
 HRESULT [Successors](#) ([out, retval] [IOrgObjects](#) \*\*pVal)  
 HRESULT [Attributes](#) ([out, retval] [IOrgAttributes](#) \*\*pVal)  
 HRESULT [GetPredecessorsByClassName](#) ([in] BSTR bstrClassName,[in]  
 VARIANT\_BOOL bDirectOnly,[out, retval] [IOrgObjects](#) \*\*pList)  
 HRESULT [GetSuccessorsByClassName](#) ([in]BSTR bstrClassId,[out, retval]  
[IOrgObjects](#) \*\*SuccList)  
 HRESULT [GetPredecessorsByName](#) ([in] BSTR bstrName,[out, retval] [IOrgObjects](#)  
 \*\*pList)  
 HRESULT [GetSuccessorsByName](#) ([in]BSTR bstrName,[out, retval] [IOrgObjects](#)  
 \*\*pList)  
 HRESULT [Substitutes](#) ([out, retval][IOrgObjects](#) \*\*pVal)

## Dokumentation der Elementfunktionen

HRESULT Id ([out, retval] BSTR \* *pVal*)  
**Beschreibung:** Diese Methode liefert die Objektid.

### Rückgabewerte:

*BSTR*: ObjektId

### Beispiel:

```

Dim MyObjectName ` Name des Organisationsobjektes

Set ListOrgObject = wfororganisation.GetOrgObjectsByName(MyObjectName)

If ListOrgObject.count >= 1 Then
    `Liste enthält mindestens ein Objekt, vom ersten wird Id
    `ermittelt
    ObjectId = ListOrgObject.item(0).Id
End If
  
```

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

**Beschreibung:** Diese Methode liefert den im **enaio®** - Workflow Editor definierten Objektnamen.

**Rückgabewerte:**

*BSTR*: Objektname

**Beispiel:**

```
Dim MyObjectId ` Id des Organisationsobjektes

Set MyOrgObject = wfororganisation.GetOrgObjectById(MyObjectId)

ObjName = MyOrgObject.Name
```

HRESULT Predecessors ([out, retval] [IOrgObjects](#) \*\* *pVal*)

**Beschreibung:** Diese Methode liefert eine Liste aller direkten Vorgängerobjekte aus der Organisations-struktur.

**Rückgabewerte:**

*IOrgObjects*: Liste aller Vorgängerobjekte

**Beispiel:**

```
Dim MyObjectId ` Id des Organisationsobjektes

Set predecessors =
wfororganisation.GetOrgObjectById(MyObjectId).Predecessors

`Liste enthält mindestens ein Vorgänger, vom ersten wird der Name
ermittelt
If predecessors.count >= 1 Then
    PredecessorName = predecessors.item(0).Name
End If
```

HRESULT Successors ([out, retval] [IOrgObjects](#) \*\* *pVal*)

**Beschreibung:** Diese Methode liefert eine Liste aller direkten Nachfolgerobjekte aus der Organisations-struktur.

**Rückgabewerte:**

*IOrgObjects*: Liste aller Nachfolgerobjekte

**Beispiel:** siehe [Predecessors](#)

HRESULT Attributes ([out, retval] [IOrgAttributes](#) \*\* pVal)

**Beschreibung:** Diese Methode liefert eine Liste aller Attribute einer Organisationsobjektes.

**Rückgabewerte:**

*IOrgObjAttributes:* Liste der Organisationsobjektattribute

**Beispiel:** Es wird der Vorname eines Personenobjektes ermittelt.

```
Dim MyObjectId ` Id des Organisationsobjektes

Set OrgObject = wfororganisation.GetOrgObjectById(MyObjectId)
Set personattributes = OrgObject.Attributes

For i = 0 To personattributes.count - 1
    If personattributes.item(i).name = "Vorname" Then
        Firstname = personattributes.item(i).value
        Exit For
    End If
Next
```

HRESULT GetPredecessorsByClassName ([in] BSTR *bstrClassName*, [in] VARIANT\_BOOL *bDirectOnly*, [out, retval] [IOrgObjects](#) \*\* *pList*)

**Beschreibung:** Diese Methode liefert eine Liste aller Vorgänger des Objektes anhand des Klassennamens. Der Aufruf ist mit dem Jokerzeichen '%' möglich.

**Parameter:**

*BSTR*: Name der Klasse

*VARIANT\_BOOL*: TRUE = es werden nur direkte Vorgänger zurückgeliefert, FALSE -> noch nicht implementiert

**Rückgabewerte:**

*IOrgObjects*: Liste aller Vorgängerobjekte

**Beispiel:** Es wird der Name eines Vorgängers ermittelt.

```
Dim MyObjectId    ` Id des Organisationsobjektes
Dim Classname     ` Name der Organisationsklasse

`Organisationsobjekt erstellen
Set OrgObject = wforganisation.GetOrgObjectById(MyObjectId)

`alle direkten Vorgänger ermitteln
Set predecessors = OrgObject.GetPredecessorsByClassName(Classname, _
True)

If predecessors.count >= 1 Then
    `Name des ersten Vorgängers ermitteln
    PredecessorName = predecessors.item(0).Name
End If
```

HRESULT GetSuccessorsByClassName ([in] BSTR *bstrClassId*, [out, retval] [IOrgObjects](#) \*\* *SuccList*)

**Beschreibung:** Diese Methode liefert eine Liste aller direkten Nachfolger anhand des Klassennamens. Der Aufruf ist mit dem Jokerzeichen '%' möglich.

**Parameter:**

*BSTR*: Name der Klasse

**Rückgabewerte:**

*IOrgObjects*: Liste aller Nachfolgerobjekte

**Beispiel:** siehe [GetPredecessorsByClassName](#)

HRESULT GetPredecessorsByName ([in] BSTR *bstrName*, [out, retval] [IOrgObjects](#) \*\*  
*pList*)

**Beschreibung:** Diese Methode liefert eine Liste aller direkten Vorgänger anhand des Objektnamens. Der Aufruf ist mit dem Jokerzeichen '%' möglich.

**Parameter:**

*BSTR*: Name des Objektes

**Rückgabewerte:**

*IOrgObjects*: Liste aller Nachfolgerobjekte

**Beispiel:** siehe [GetPredecessorsByClassName](#)

HRESULT GetSuccessorsByName ([in] BSTR *bstrName*, [out, retval] [IOrgObjects](#) \*\*  
*pList*)

**Beschreibung:** Diese Methode liefert eine Liste aller direkten Nachfolger anhand des Objektnamens. Der Aufruf ist mit dem Jokerzeichen '%' möglich.

**Parameter:**

*BSTR*: Name des Objektes

**Rückgabewerte:**

*IOrgObjects*: Liste aller Nachfolgerobjekte

**Beispiel:** siehe [GetPredecessorsByClassName](#)

HRESULT Substitutes ([out, retval] [IOrgObjects](#) \*\* *pVal*)

**Beschreibung:** Diese Methode ermittelt zu einer Person oder einer Rolle die Stellvertreter.

**Rückgabewerte:**

*IOrgObjects*: Liste aller Stellvertreter

**Beispiel:**

```
Dim MyObjectId ' Id des Organisationsobjektes

'Organisationsobjekt erstellen
Set OrgObject = wforganisation.GetOrgObjectById(MyObjectId)

'alle Vertreter ermitteln
Set substitutes = OrgObject.Substitutes
```



## IOrgObjects Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Liste aller Organisationsobjekte. Diese Schnittstelle ist von [IComCollection](#) abgeleitet.

**Zugriff auf die Liste:** siehe [IOrganisation](#) und [IOrgObject](#)

**Elemente der Liste:** siehe [IOrgObject](#)

## IRoutingLists Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle ist von IComCollection abgeleitet und enthält Elemente vom Typ IRoutingList.

**Zugriff auf die Liste:** siehe IWFCClient

**Siehe auch:** [IRoutingList](#), [IRoutingListEntry](#), [IRoutingListItem](#)

## IRoutingList Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt die Abarbeitungsliste einer Adhocaktivität. Diese Schnittstelle ist von [IComCollection](#) abgeleitet und enthält Elemente vom Typ [IRoutingListEntry](#).

**Zugriff auf die Liste:** siehe [IActiveWorkItem](#)

**Elemente der Liste:** siehe [IRoutingListEntry](#)

### Öffentliche Methoden

HRESULT [ID](#)([out, retval] BSTR\* pVal)  
 HRESULT [CreateNewEntry](#)([out, retval] [IRoutingListEntry](#)\*\* pEntry)  
 HRESULT [CreateNewItem](#)([in] IWFPProcessActivity, [out, retval] [IRoutingListItem](#)\*\* pItem)  
 HRESULT [AdhocModActivityId](#)([out, retval] BSTR\* pVal)  
 HRESULT [LegalItemActivities](#)([out, retval] IWFPProcessActivities\*\* pVal)  
 HRESULT [ReminderPeriodTemplates](#) ([out, retval] IAdhocReminderPeriodTemplates\*\* pVal)  
 HRESULT [IsTemplate](#)([out, retval] VARIANT\_BOOL\* pVal)  
 HRESULT [TemplateID](#)([out, retval] BSTR\* pVal)  
 HRESULT [TemplateName](#)([out, retval] BSTR\* pVal)  
 HRESULT [TemplateIsPublic](#)([out, retval] VARIANT\_BOOL\* pVal)  
 HRESULT [Remark](#)([out, retval] BSTR\* pVal)  
 HRESULT [CreateNewTemplate](#)([in] BSTR templateName, [in] VARIANT\_BOOL bIsPublic, [out, retval] BSTR\* pTemplateId)  
 HRESULT [UpdateTemplate](#)([in] BSTR templateId, [in] BSTR templateName, [in] VARIANT\_BOOL bIsPublic)  
 HRESULT [InsertTemplate](#)([in] [IRoutingList](#)\* pTemplate, [in] long insertionIndex);  
 HRESULT [CopyItem](#)([in] [IRoutingListItem](#)\* pSourceItem, [out, retval] [IRoutingListItem](#)\*\* pDestItem)  
 HRESULT [PublishTemplate](#)()  
 HRESULT [UnpublishTemplate](#)()  
 HRESULT [IsExpandable](#)([out, retval] VARIANT\_BOOL\* pbVal)  
 HRESULT [IsExpandable](#) ([in] VARIANT\_BOOL bVal)  
 HRESULT [IsValid](#)([out, retval] VARIANT\_BOOL \* pbVal)

### Dokumentation der Elementfunktionen

HRESULT ID ([out, retval] BSTR \* pVal)

**Beschreibung:** Diese Methode liefert die Id der Abarbeitungsliste.

**Rückgabewerte:**

*BSTR*: Id der Abarbeitungsliste

**Beispiel:**

```
Rid = workitem.RoutingList.ID
```

HRESULT CreateNewEntry([out, retval] [IRoutingListEntry](#)\*\* pEntry)

**Beschreibung:** Diese Methode erzeugt einen neuen Eintrag für die Abarbeitungsliste, fügt ihn aber nicht ein.

**Rückgabewerte:**

[IRoutingListEntry](#): Neuer Eintrag

**Beispiel:**

```
Set newEntry = workitem.RoutingList.CreateNewEntry
```

HRESULT CreateNewItem([in] IWFPProcessActivity, [out, retval] [IRoutingListItem](#)\*\* pItem)

**Beschreibung:** Diese Methode erzeugt ein neues Element für einen Eintrag der Abarbeitungsliste, fügt es aber nicht ein.

**Parameter:**

IWFPProcessActivity: Aktivität im Modell, für die ein Adhoc-Element erzeugt werden soll.

**Rückgabewerte:**

[IRoutingListItem](#): Neues Element

**Beispiel:**

```
Set LegalItemActivities = workitem.RoutingList.LegalItemActivities
Set newItem =
workitem.RoutingList.CreateNewItem(LegalItemActivities.Item(0))
```

HRESULT AdhocModActivityId([out, retval] BSTR\* pVal)

**Beschreibung:** Diese Methode liefert die ID der Adhocaktivität im Modell.

**Rückgabewerte:**

*BSTR*: Id der Adhocaktivität im Modell

**Beispiel:**

```
Aid = workitem.RoutingList.AdhocModActivityId
```

HRESULT LegalItemActivities ([out, retval] IWFPProcessActivities \*\* pVal)

**Beschreibung:** Diese Methode liefert die zulässigen Item-Aktivitäten der Routingliste.

**Rückgabewerte:**

IWFPProcessActivities: die zulässigen Item-Aktivitäten der Routinglistex

**Beispiel:**

```
Set LegalItemActivities = workitem.RoutingList.GetLegalItemActivities
```

HRESULT ReminderPeriodTemplates ([out, retval] IAdhocReminderPeriodTemplates\*\* pVal)

**Beschreibung:** Diese Methode liefert die Vorlagen für mögliche Fristen für RoutingListItems dieser RoutingListe.

**Rückgabewerte:**

ReminderPeriodTemplates\*\*.: Vorlagen für mögliche Fristen für RoutingListItems dieser RoutingListe.

**Beispiel:**

```
Set LegalItemActivities = workitem.RoutingList.LegalItemActivities
Set newItem =
workitem.RoutingList.CreateNewItem(LegalItemActivities.Item(0))
Set firstPeriodTemplate =
workitem.RoutingList.ReminderPeriodTemplates.Item(0)
newItem.ReminderPeriod = firstPeriodTemplate
```

HRESULT IsTemplate([out, retval] VARIANT\_BOOL\* pVal)

**Beschreibung:** Diese Methode gibt an, ob es sich bei diesem Objekt um eine Vorlage für eine Routingliste handelt.

**Rückgabewerte:**

VARIANT\_BOOL.: wahr, wenn es sich bei diesem Objekt um eine Vorlage für eine Routingliste handelt

HRESULT TemplateID([out, retval] BSTR\* pVal)

**Beschreibung:** Diese Methode liefert die Id der Vorlage, falls es sich bei diesem Objekt um eine Vorlage für eine Routingliste handelt.

**Rückgabewerte:**

BSTR.: Id der Vorlage, falls es sich bei diesem Objekt um eine Vorlage für eine Routingliste handelt

HRESULT TemplateName([out, retval] BSTR\* pVal)

**Beschreibung:** Diese Methode liefert den Namen der Vorlage, falls es sich bei diesem Objekt um eine Vorlage für eine Routingliste handelt.

**Rückgabewerte:**

BSTR.: Name der Vorlage, falls es sich bei diesem Objekt um eine Vorlage für eine Routingliste handelt

HRESULT TemplateIsPublic([out, retval] VARIANT\_BOOL\* pVal)

**Beschreibung:** Diese Methode gibt an, ob die Vorlage öffentlich ist, falls es sich bei diesem Objekt um eine Vorlage für eine Routingliste handelt.

**Rückgabewerte:**

VARIANT\_BOOL.: wahr, wenn die Vorlage öffentlich ist

HRESULT Remark([out, retval] BSTR\* pVal)

**Beschreibung:** Diese Methode liefert die Bemerkung zu dem momentanen Adhoc-Arbeitsschritt.

**Rückgabewerte:**

BSTR.: die Bemerkung zu dem momentanen Adhoc-Arbeitsschritt

### Beispiel:

```
Remark = workitem.RoutingList.Remark
```

HRESULT CreateNewTemplate([in] BSTR templateName, [in] VARIANT\_BOOL bIsPublic, [out, retval] BSTR\* pTemplateId)

**Beschreibung:** Diese Methode speichert die Routingliste als neue Vorlage ab.

#### Parameter:

*BSTR templateName:* Name der neuen Vorlage.

*VARIANT\_BOOL bIsPublic:* wahr, wenn die Vorlage öffentlich sein soll

#### Rückgabewerte:

*BSTR:* Id der neuen Vorlage

### Beispiel:

```
NewTemplateId = workitem.RoutingList.CreateNewTemplate "My New  
Template", false
```

HRESULT UpdateTemplate([in] BSTR templateId, [in] BSTR templateName, [in] VARIANT\_BOOL bIsPublic)

**Beschreibung:** Diese Methode speichert die Routingliste in einer bereits existierenden Vorlage ab, welche dabei überschrieben wird. Ggf. im Speicher existierende Objekte dieses Templates werden nicht aktualisiert.

#### Parameter:

*BSTR templateId:* Id der Vorlage.

*BSTR templateName:* Neuer Name der Vorlage.

*VARIANT\_BOOL bIsPublic:* wahr, wenn die Vorlage öffentlich sein soll

### Beispiel:

```
workitem.RoutingList.CreateNewTemplate  
"01234567890123456789012345678901", "My New Template", false
```

HRESULT InsertTemplate([in] [IRoutingList](#)\* pTemplate, [in] long insertionIndex)

**Beschreibung:** Diese Methode fügt eine Lauflistenvorlage in eine Laufliste ein.

#### Parameter:

*[IRoutingList](#)\* template:* die Vorlage.

*long insertionIndex:* (Entry-)Position in der Laufliste, wo die Vorlage eingefügt werden soll.

### Beispiel:

```
tid = workitem.routinglist.createNewTemplate("Meine Vorlage", false)  
Set templateList = workitem.GetAdhocTemplate(tid)  
workitem.routinglist.InsertTemplate templateList, 1
```

HRESULT CopyItem([in] [IRoutingListItem](#)\* pSourceItem, [out, retval] [IRoutingListItem](#)\*\* pDestItem)

**Beschreibung:** Diese Methode erzeugt ein neues Item für die Routingliste, indem es ein Item einer anderen Routingliste (oder eines Templates) kopiert.

**Parameter:**

[IRoutingListItem](#)\* pSourceItem: das zu kopierende Item.

**Rückgabewerte:**

[IRoutingListItem](#)\*: das kopierte Item

HRESULT PublishTemplate()

**Beschreibung:** Diese Methode veröffentlicht dieses Template (funktioniert nur für Templates).

HRESULT UnpublishTemplate()

**Beschreibung:** Diese Methode privatisiert dieses Template für den angemeldeten Benutzer (funktioniert nur für Templates).

HRESULT IsExpandable ([out, retval] VARIANT\_BOOL\* pbVal)

**Beschreibung:** Diese Methode gibt an, ob in dieser Liste weitere Entries eingefügt werden dürfen.

**Rückgabewerte:**

VARIANT\_BOOL: true, wenn in dieser Liste weitere Entries eingefügt werden dürfen.

HRESULT IsExpandable ([in] VARIANT\_BOOL bVal)

**Beschreibung:** Diese Methode setzt, ob in dieser Liste weitere Entries eingefügt werden dürfen. Darf nur durch den Prozessverantwortlichen aufgerufen werden.

**Parameter:**

*VARIANT\_BOOL*: true, wenn in dieser Liste weitere Entries eingefügt werden dürfen

HRESULT IsValid([out, retval] VARIANT\_BOOL \* pbVal)

**Beschreibung:** Diese Methode gibt an, ob sämtliche Items in der Routingliste zulässig sind.

**Rückgabewerte:**

VARIANT\_BOOL: true, wenn sämtliche Items in der Routingliste zulässig sind.

## IRoutingListEntry Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt einen Eintrag in der Abarbeitungsliste einer Adhocaktivität. Diese Schnittstelle ist von [IComCollection](#) abgeleitet und enthält Elemente vom Typ [IRoutingListItem](#).

**Zugriff auf die Liste:** siehe [IRoutingList](#)

**Elemente der Liste:** siehe [IRoutingListItem](#)

### Öffentliche Methoden

HRESULT [IsExpandable](#) ([out, retval] VARIANT\_BOOL\* pbVal)

HRESULT [IsExpandable](#) ([in] VARIANT\_BOOL bVal)

HRESULT [IsFlagChangeByProcessResponsibleOnly](#) ([out, retval] VARIANT\_BOOL\* pbVal)

### Dokumentation der Elementfunktionen

HRESULT IsExpandable ([out, retval] VARIANT\_BOOL\* pbVal)

**Beschreibung:** Diese Methode gibt an, ob in diesem Entry weitere Items eingefügt werden dürfen.

**Rückgabewerte:**

VARIANT\_BOOL: true, wenn in diesem Entry weitere Items eingefügt werden dürfen.

HRESULT IsExpandable ([in] VARIANT\_BOOL bVal)

**Beschreibung:** Diese Methode setzt, ob in diesem Entry weitere Items eingefügt werden dürfen.

**Parameter:**

*VARIANT\_BOOL:* true, wenn in diesem Entry weitere Items eingefügt werden dürfen

HRESULT IsFlagChangeByProcessResponsibleOnly ([out, retval] VARIANT\_BOOL\* pbVal)

**Beschreibung:** Diese Methode gibt an, ob das Flag [IsExpandable](#) dieses Entrys nur vom Prozessverantwortlichen verändert werden darf. Im Allgemeinen darf das Flag bei neu angelegten Entries von dem aktuell angemeldeten Benutzer auch geändert werden, wenn dieser nicht Prozessverantwortlicher ist. Bei Entries, die dem Server bereits bekannt sind, darf dies nur der Prozessverantwortliche.

**Rückgabewerte:**

VARIANT\_BOOL: true, wenn das Flag IsExpandable dieses Entrys nur vom Prozessverantwortlichen verändert werden darf.



## IRoutingListItem Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt ein Element eines Eintrags in der Abarbeitungsliste einer Adhocaktivität.

**Zugriff:** siehe [IRoutingListEntry](#)

### Öffentliche Methoden

```

HRESULT ID([out, retval] BSTR* pVal)
HRESULT Activity ([in] IWFProcessActivity* pActivity)
HRESULT ActivityId([out, retval] BSTR* pVal)
HRESULT ActivityName([out, retval] BSTR* pVal)
HRESULT ActivityName ([in] BSTR newVal)
HRESULT Remark ([out, retval] BSTR* newVal)
HRESULT Remark ([in] BSTR newVal)
HRESULT ModelActivityName([out, retval] BSTR* pVal)
HRESULT ReminderPeriodId([out, retval] BSTR* pVal)
HRESULT ReminderPeriod([in] IAdhocReminderPeriodTemplate newVal)
HRESULT ReminderPeriodName([out, retval] BSTR* pVal)
HRESULT ReminderPeriodDuration ([out, retval] long* pVal)
HRESULT ReminderPeriodDuration ([in] long newVal)
HRESULT ReminderPeriodIsAbsDuration ([out, retval] long* pVal)
HRESULT ReminderPeriodIsAbsDuration ([in] long newVal)
HRESULT ReminderPeriodTemplates ([out, retval] IAdhocReminderPeriodTemplates** pVal)
HRESULT Participants ([out, retval] IRoutingListItemParticipants** pVal)
HRESULT IsDeleteable([out, retval] VARIANT_BOOL* pbVal)
HRESULT IsDeleteable([in] VARIANT_BOOL bVal)
HRESULT IsChangeable([out, retval] VARIANT_BOOL* pbVal)
HRESULT IsChangeable([in] VARIANT_BOOL bVal)
HRESULT IsFlagChangeByProcessResponsibleOnly ([out, retval] VARIANT_BOOL* pbVal)
HRESULT Validate([out, retval] long* pVal)

```

### Dokumentation der Elementfunktionen

HRESULT ID ([out, retval] BSTR \* pVal)

**Beschreibung:** Diese Methode liefert die Id des Elements der Abarbeitungsliste.

**Rückgabewerte:**

*BSTR*: Id des Elements der Abarbeitungsliste

**Beispiel:**

```

Set LegalItemActivities = workitem.RoutingList.LegalItemActivities
Set LegalItemActivities = workitem.RoutingList.LegalItemActivities
Set newItem =
workitem.RoutingList.CreateNewItem(LegalItemActivities.Item(0))
IID = newItem.ID

```

HRESULT Activity ([in] IWFProcessActivity\* pActivity)

**Beschreibung:** Diese Methode setzt die Aktivität dieses Elements der Abarbeitungsliste.

**Parameter:**

*IWFProcessActivity*: Aktivität dieses Elements der Abarbeitungsliste.

HRESULT ActivityId ([out, retval] BSTR \* pVal)

**Beschreibung:** Diese Methode liefert die Id der Aktivität im Modell dieses Elements der Abarbeitungsliste.

**Rückgabewerte:**

*BSTR*: Id der Aktivität im Modell dieses Elements der Abarbeitungsliste.

**Beispiel:**

```
Set LegalItemActivities = workitem.RoutingList.LegalItemActivities
Set newItem =
workitem.RoutingList.CreateNewItem(LegalItemActivities.Item(0))
AId = newItem.ActivityId
```

HRESULT ActivityName ([out, retval] BSTR \* pVal)

**Beschreibung:** Diese Methode liefert den Namen der Aktivität dieses Elements der Abarbeitungsliste.

**Rückgabewerte:**

*BSTR*: Name der Aktivität dieses Elements der Abarbeitungsliste.

**Beispiel:**

```
Set LegalItemActivities = workitem.RoutingList.LegalItemActivities
Set newItem =
workitem.RoutingList.CreateNewItem(LegalItemActivities.Item(0))
Name = newItem.ActivityName
```

HRESULT ActivityName ([in] BSTR newVal)

**Beschreibung:** Diese Methode setzt den Namen der Aktivität dieses Elements der Abarbeitungsliste.

**Parameter:**

*BSTR*: neuer Aktivitätsname.

**Beispiel:**

```
Set LegalItemActivities = workitem.RoutingList.LegalItemActivities
Set newItem =
workitem.RoutingList.CreateNewItem(LegalItemActivities.Item(0))
newItem.ActivityName = "Kenntnisnahme"
```

HRESULT Remark ([out, retval] BSTR\* newVal)

**Beschreibung:** Diese Methode liefert die Bemerkung zu diesem Element der Abarbeitungsliste.

**Rückgabewerte:**

*BSTR*: Bemerkung zu diesem Element der Abarbeitungsliste.

**Beispiel:**

```
Set LegalItemActivities = workitem.RoutingList.LegalItemActivities
Set newItem =
workitem.RoutingList.CreateNewItem(LegalItemActivities.Item(0))
Remark = newItem.Remark
```

HRESULT Remark ([in] BSTR newVal)

**Beschreibung:** Diese Methode setzt die Bemerkung zu diesem Element der Abarbeitungsliste.

**Parameter:**

*BSTR*: neue Bemerkung.

**Beispiel:**

```
Set LegalItemActivities = workitem.RoutingList.LegalItemActivities
Set newItem =
workitem.RoutingList.CreateNewItem(LegalItemActivities.Item(0))
newItem.Remark = "Bitte anschauen und abhaken!"
```

HRESULT ModelActivityName ([out, retval] BSTR \* pVal)

**Beschreibung:** Diese Methode liefert den Modellnamen der Aktivität dieses Elements der Abarbeitungsliste.

**Rückgabewerte:**

*BSTR*: Modellname der Aktivität dieses Elements der Abarbeitungsliste.

**Beispiel:**

```
Set LegalItemActivities = workitem.RoutingList.LegalItemActivities
Set newItem =
workitem.RoutingList.CreateNewItem(LegalItemActivities.Item(0))
Name = newItem.ModelActivityName
```

HRESULT ReminderPeriodId ([out, retval] BSTR \* pVal)

**Beschreibung:** Diese Methode liefert die Id der Mahnfristenvorlage für dieses Element der Abarbeitungsliste.

**Rückgabewerte:**

*BSTR*: Id der Mahnfristenvorlage für dieses Element der Abarbeitungsliste.

**Beispiel:**

```
Set LegalItemActivities = workitem.RoutingList.LegalItemActivities
Set newItem =
workitem.RoutingList.CreateNewItem(LegalItemActivities.Item(0))
Set firstPeriodTemplate =
workitem.RoutingList.ReminderPeriodTemplates.Item(0)
newItem.ReminderPeriod = firstPeriodTemplate
Rid = newItem.ReminderPeriodId
```

HRESULT ReminderPeriod ([in] [IAdhocReminderPeriodTemplate](#) period)

**Beschreibung:** Diese Methode setzt die Mahnfristenvorlage für dieses Element der Abarbeitungsliste. Intern wird auch der Vorlage entsprechend der Name und die Dauer der Frist gesetzt.

**Parameter:**

*BSTR*: die Mahnfristenvorlage für dieses Element der Abarbeitungsliste

**Beispiel:**

```
Set LegalItemActivities = workitem.RoutingList.LegalItemActivities
Set newItem =
workitem.RoutingList.CreateNewItem(LegalItemActivities.Item(0))
Set firstPeriodTemplate =
workitem.RoutingList.ReminderPeriodTemplates.Item(0)
newItem.ReminderPeriod = firstPeriodTemplate
Rid = newItem.ReminderPeriodId
```

HRESULT ReminderPeriodName ([out, retval] BSTR \* pVal)

**Beschreibung:** Diese Methode liefert den Namen der Mahnfristenvorlage für dieses Element der Abarbeitungsliste.

**Rückgabewerte:**

*BSTR*: Name der Mahnfristenvorlage für dieses Element der Abarbeitungsliste.

**Beispiel:**

```
Set LegalItemActivities = workitem.RoutingList.LegalItemActivities
Set newItem =
workitem.RoutingList.CreateNewItem(LegalItemActivities.Item(0))
Set firstPeriodTemplate =
workitem.RoutingList.ReminderPeriodTemplates.Item(0)
newItem.ReminderPeriod = firstPeriodTemplate
RName = newItem.ReminderPeriodName
```

HRESULT ReminderPeriodDuration ([out, retval] long \* pVal)

**Beschreibung:** Diese Methode liefert die Dauer der eingerichteten Frist in Sekunden. Je nach dem wie ReminderPeriodIsAbsDuration eingerichtet ist, ist dies ein absoluter oder relativer Wert.

**Rückgabewerte:**

*long*: die Dauer der eingerichteten Frist

**Beispiel:**

```
Set LegalItemActivities = workitem.RoutingList.LegalItemActivities
Set newItem =
workitem.RoutingList.CreateNewItem(LegalItemActivities.Item(0))
Set firstPeriodTemplate =
workitem.RoutingList.ReminderPeriodTemplates.Item(0)
newItem.ReminderPeriod = firstPeriodTemplate
duration = newItem.ReminderPeriodDuration
```

HRESULT ReminderPeriodDuration([in] long newVal)

**Beschreibung:** Diese Methode setzt die Dauer der eingerichteten Frist in Sekunden. Je nach dem wie ReminderPeriodIsAbsDuration eingerichtet ist, ist dies ein absoluter oder relativer Wert.

**Parameter:**

*long*: Dauer der eingerichteten Frist in Sekunden

**Beispiel:**

```
Set LegalItemActivities = workitem.RoutingList.LegalItemActivities
Set newItem =
workitem.RoutingList.CreateNewItem(LegalItemActivities.Item(0))
Set firstPeriodTemplate =
workitem.RoutingList.ReminderPeriodTemplates.Item(0)
newItem.ReminderPeriod = firstPeriodTemplate
newItem.ReminderPeriodDuration = 1207044000
newItem.ReminderPeriodIsAbsDuration = true
```

HRESULT ReminderPeriodIsAbsDuration ([out, retval] VARIANT\_BOOL \* pVal)

**Beschreibung:** Diese Methode gibt an, ob die konfigurierte Dauer der Frist ein absoluter oder ein relativer Wert ist.

**Rückgabewerte:**

*VARIANT\_BOOL*: true, wenn die Dauer der Frist ein absoluter Wert ist; sonst false.

**Beispiel:**

```
Set LegalItemActivities = workitem.RoutingList.LegalItemActivities
Set newItem =
workitem.RoutingList.CreateNewItem(LegalItemActivities.Item(0))
Set firstPeriodTemplate =
workitem.RoutingList.ReminderPeriodTemplates.Item(0)
IsAbs = newItem.ReminderPeriodIsAbsDuration
```

HRESULT ReminderPeriodIsAbsDuration([in] VARIANT\_BOOL newVal)

**Beschreibung:** Diese Methode setzt, ob die konfigurierte Dauer der Frist ein absoluter oder ein relativer Wert ist.

**Parameter:**

*VARIANT\_BOOL*: true, wenn die Dauer der Frist ein absoluter Wert ist; sonst false.

**Beispiel:**

```
Set LegalItemActivities = workitem.RoutingList.LegalItemActivities
Set newItem =
workitem.RoutingList.CreateNewItem(LegalItemActivities.Item(0))
Set firstPeriodTemplate =
workitem.RoutingList.ReminderPeriodTemplates.Item(0)
newItem.ReminderPeriod = firstPeriodTemplate
newItem.ReminderPeriodDuration = 1207044000
newItem.ReminderPeriodIsAbsDuration = true
```

HRESULT ReminderPeriodTemplates ([out, retval] IAdhocReminderPeriodTemplates\*\* pVal)

**Beschreibung:** Diese Methode liefert die Vorlagen für mögliche Fristen des RoutingListItems.

**Rückgabewerte:**

[ReminderPeriodTemplates](#)\*: Vorlagen für mögliche Fristen des RoutingListItems.

**Beispiel:**

```
Set LegalItemActivities = workitem.RoutingList.LegalItemActivities
Set newItem =
workitem.RoutingList.CreateNewItem(LegalItemActivities.Item(0))
Set firstPeriodTemplate = newItem.ReminderPeriodTemplates.Item(0)
newItem.ReminderPeriod = firstPeriodTemplate
```

HRESULT Participants ([out, retval] [IRoutingListItemParticipants](#)\*\* pVal)

**Beschreibung:** Diese Methode liefert die Teilnehmer des RoutingListItems als veränderbare Liste von [IOrgObject](#).

**Rückgabewerte:**

[IRoutingListItemParticipants](#)\*: die Teilnehmer des RoutingListItems als veränderbare Liste von [IOrgObject](#).

**Beispiel:**

```
Set LegalItemActivities = workitem.RoutingList.LegalItemActivities
Set newItem =
workitem.RoutingList.CreateNewItem(LegalItemActivities.Item(0))
Set participants = newItem.Participants
Participants.add 0, wforganisation.GetOrgObjectByName("RoleX").Item(0)
```

HRESULT IsDeleteable([out, retval] VARIANT\_BOOL\* pbVal)

**Beschreibung:** Diese Methode gibt an, ob das RoutingListItem löscherbar ist.

**Rückgabewerte:**

VARIANT\_BOOL: true, wenn das RoutingListItem löscherbar ist.

HRESULT IsDeleteable([in] VARIANT\_BOOL bVal)

**Beschreibung:** Diese Methode setzt, ob das RoutingListItem löscherbar ist.

**Parameter:**

*VARIANT\_BOOL*: true, wenn das RoutingListItem löscherbar ist

HRESULT IsChangeable([out, retval] VARIANT\_BOOL\* pbVal)

**Beschreibung:** Diese Methode gibt an, ob das RoutingListItem änderbar ist.

**Rückgabewerte:**

VARIANT\_BOOL: true, wenn das RoutingListItem änderbar ist.

HRESULT IsChangeable([in] VARIANT\_BOOL bVal)

**Beschreibung:** Diese Methode setzt, ob das RoutingListItem änderbar ist.

**Parameter:**

*VARIANT\_BOOL*: true, wenn das RoutingListItem änderbar ist

HRESULT IsFlagChangeByProcessResponsibleOnly([out, retval] VARIANT\_BOOL\* pbVal)

**Beschreibung:** Diese Methode gibt an, ob die Flags [IsDeleteable](#) und [IsChangeable](#) dieses Items nur vom Prozessverantwortlichen verändert werden dürfen. Im Allgemeinen dürfen die Flags bei neu angelegten Items von dem aktuell angemeldeten Benutzer auch geändert werden, wenn dieser nicht Prozessverantwortlicher ist. Bei Items, die dem Server bereits bekannt sind, darf dies nur der Prozessverantwortliche.

**Rückgabewerte:**

*VARIANT\_BOOL*: true, wenn die Flags [IsDeleteable](#) und [IsChangeable](#) dieses Items nur vom Prozessverantwortlichen verändert werden dürfen.

HRESULT Validate([out, retval] long\* pVal)

**Beschreibung:** Diese Methode prüft, ob das Item für die entsprechende Laufliste zulässig ist.

**Rückgabewerte:**

long: Flags (sind keine Flags gesetzt, also Wert = 0, dann ist das Item zulässig):

0x00000001: Aktivität ist nicht zulässig

0x00000002: Frist ist nicht zulässig

## IRoutingListItemParticipants Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Menge von Teilnehmern für ein Item in einer Routingliste. Diese Schnittstelle ist von [IComCollection](#) abgeleitet und enthält Elemente vom Typ IOrgObject. Beim Ändern der Liste (Einfügen / Entfernen von Teilnehmern) wird geprüft, ob dies für das entsprechende Item überhaupt zulässig ist.

**Zugriff auf die Liste:** siehe [IRoutingListItem](#)



## IAdhocReminderPeriodTemplate Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Vorlage für eine Mahnfrist, die für RoutingListItems verwendet werden kann.

**Zugriff:** siehe [IRoutingListItem](#), [IRoutingList](#)

### Öffentliche Methoden

HRESULT [ID](#)([out, retval] BSTR\* pVal)  
HRESULT [Name](#)([out, retval] BSTR\* pVal)  
HRESULT [DefaultDuration](#)([out, retval] long\* pVal)

### Dokumentation der Elementfunktionen

HRESULT ID ([out, retval] BSTR \* pVal)

**Beschreibung:** Diese Methode liefert die Id der Mahnfristenvorlage.

**Rückgabewerte:**

*BSTR*: Id der Mahnfristenvorlage.

**Beispiel:**

```
Set firstPeriodTemplate =  
workitem.RoutingList.ReminderPeriodTemplates.Item(0)  
TId = firstPeriodTemplate.ID
```

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

**Beschreibung:** Diese Methode liefert den Namen der Mahnfristenvorlage.

**Rückgabewerte:**

*BSTR*: Name der Mahnfristenvorlage.

**Beispiel:**

```
Set firstPeriodTemplate =  
workitem.RoutingList.ReminderPeriodTemplates.Item(0)  
TName = firstPeriodTemplate.Name
```

HRESULT DefaultDuration ([out, retval] long \* pDefaultDuration)

**Beschreibung:** Diese Methode liefert die voreingestellte Dauer (relative Zeitangabe im Sekunden) der Mahnfristenvorlage.

**Rückgabewerte:**

*LONG*: voreingestellte Dauer (relative Zeitangabe im Sekunden) der Mahnfristenvorlage.

**Beispiel:**

```
Set firstPeriodTemplate =  
workitem.RoutingList.ReminderPeriodTemplates.Item(0)  
TDuration = firstPeriodTemplate.DefaultDuration
```

## IAdhocReminderPeriodTemplates Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Liste mit Vorlagen für Mahnfristen, die für `RoutingListItems` verwendet werden kann.

**Elemente der Liste:** [IAdhocReminderPeriodTemplate](#)

## IComRoutingList Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt die Abarbeitungsliste einer Ad-hoc-aktivität. Diese Schnittstelle ist von [IComCollection](#) abgeleitet und enthält Elemente vom Typ [IComRoutingListEntry](#).

**Zugriff auf die Liste:** siehe [IComCollection](#)

**Elemente der Liste:** siehe [IComRoutingListEntry](#)

### Öffentliche Methoden

HRESULT [CreateNewEntry](#)([out, retval] [IComRoutingListEntry](#)\*\* pEntry)  
HRESULT [CreateNewItem](#)([out, retval] [IComRoutingListItem](#)\*\* pItem)  
HRESULT Add ([in]VARIANT varItem, [in] [IComRoutingListEntry](#)\* pEntry)  
HRESULT [LegalItemActivities](#)([out, retval] [IComModelActivitiyList](#)\*\* pVal)  
HRESULT [ReminderPeriodTemplates](#) ([out, retval] [IComAdhocReminderPeriodTemplates](#)\*\* pVal)  
HRESULT [InsertTemplate](#)([in] [IComRoutingList](#)\* pTemplate, [in] long insertionIndex)

### Dokumentation der Elementfunktionen

HRESULT CreateNewEntry([out, retval] [IComRoutingListEntry](#)\*\* pEntry)

**Beschreibung:** Diese Methode erzeugt einen neuen Eintrag für die Abarbeitungsliste, fügt ihn aber nicht ein.

**Rückgabewerte:**

[IComRoutingListEntry](#): Neuer Eintrag

**Beispiel:**

```
Set newEntry = thisprocess.currentactivity.RoutingList.CreateNewEntry
```

HRESULT CreateNewItem([out, retval] [IComRoutingListItem](#)\*\* pItem)

**Beschreibung:** Diese Methode erzeugt ein neues Element für einen Eintrag der Abarbeitungsliste, fügt es aber nicht ein.

**Rückgabewerte:**

[IComRoutingListItem](#): Neues Element

**Beispiel:**

```
Set myRoutingList = thisprocess.currentactivity.RoutingList  
Set LegalItemActivities = myRoutingList.LegalItemActivities  
Set newItem = myRoutingList.CreateNewItem(LegalItemActivities.Item(0))
```

HRESULT LegalItemActivities ([out, retval] [IComModelActivitiyList](#)\*\* pVal \*\* pVal)

**Beschreibung:** Diese Methode liefert die zulässigen Item-Aktivitäten der Routingliste.

**Rückgabewerte:**

[IComModelActivitiyList](#) : die zulässigen Item-Aktivitäten der Routinglistex

**Beispiel:**

```
Set LegalItemActivities =
thisprocess.currentactivity.RoutingList.GetLegalItemActivities
```

HRESULT ReminderPeriodTemplates ([out, retval] [IComAdhocReminderPeriodTemplates](#)\*\* pVal)

**Beschreibung:** Diese Methode liefert die Vorlagen für mögliche Fristen für RoutingListItem Elemente dieser Laufliste.

**Rückgabewerte:**

[ReminderPeriodTemplates](#)\*\* : Vorlagen für mögliche Fristen für RoutingListItem Elemente dieser Laufliste.

**Beispiel:**

```
Set LegalItemActivities = workitem.RoutingList.LegalItemActivities
Set newItem =
workitem.RoutingList.CreateNewItem(LegalItemActivities.Item(0))
Set firstPeriodTemplate =
workitem.RoutingList.ReminderPeriodTemplates.Item(0)
newItem.ReminderPeriod = firstPeriodTemplate
```

HRESULT [InsertTemplate](#)([in] [IComRoutingList](#)\* pTemplate, [in] long insertionIndex)

**Beschreibung:** Diese Methode fügt eine Lauflistenvorlage in eine Laufliste ein.

**Parameter:**

[IComRoutingList](#)\* *template*: die Vorlage.

*long insertionIndex*: (Entry-)Position in der Laufliste, wo die Vorlage eingefügt werden soll.

**Beispiel:**

```
Set template = wfsystem.GetRoutingListTemplate("4711")
thisprocess.currentactivity.routinglist.InsertTemplate(template,2)
```

HRESULT Add ([in]VARIANT varItem, [in] [IComRoutingListEntry](#)\* pEntry)

**Beschreibung:** Diese Methode fügt ein IComRoutingListEntry Element in die Laufliste ein.

**Parameter:**

VARIANT varItem: bestimmt die Position des IComRoutingListEntry Elements in der Laufliste.  
[IComRoutingListEntry](#)\* *pEntry*: das einzufügende Element.

HRESULT Remove ([in]VARIANT varItem)

**Beschreibung:** Diese Methode entfernt IComRoutingListEntry Element aus der Laufliste.

**Parameter:**

VARIANT varItem: bestimmt die Position des zu löschenden Eleme IComRoutingListEntry Elements in der Laufliste

## IComRoutingListEntry Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt einen Eintrag in der Abarbeitungsliste einer Adhocaktivität. Diese Schnittstelle ist von [IComCollection](#) abgeleitet und enthält Elemente vom Typ [IComRoutingList](#).

**Zugriff auf die Liste:** siehe [IComRoutingList](#)

**Elemente der Liste:** siehe [IComRoutingListItem](#)

### Öffentliche Methoden

```
HRESULT IsExpandable ([out, retval] VARIANT_BOOL* pbVal)
HRESULT IsExpandable ([in] VARIANT_BOOL bVal)
HRESULT CreateNewItem ([out, retval] IComRoutingListItem ** pVal)
HRESULT Add ([in] VARIANT varItem, [in] IComRoutingListItem * pVal)
HRESULT Remove ([in] VARIANT varItem)
```

### Dokumentation der Elementfunktionen

```
HRESULT IsExpandable ([out, retval] VARIANT_BOOL* pbVal)
```

**Beschreibung:** Diese Methode gibt an, ob in diesem Entry weitere Items eingefügt werden dürfen.

**Rückgabewerte:**

VARIANT\_BOOL: true, wenn in diesem Entry weitere Items eingefügt werden dürfen.

```
HRESULT IsExpandable ([in] VARIANT_BOOL bVal)
```

**Beschreibung:** Diese Methode setzt, ob in diesem Entry weitere Items eingefügt werden dürfen.

**Parameter:**

*VARIANT\_BOOL*: true, wenn in diesem Entry weitere Items eingefügt werden dürfen

```
HRESULT CreateNewItem ([out, retval] IComRoutingListItem ** pVal)
```

**Beschreibung:** Erstellt ein neues Element vom Typ [IComRoutingListItem](#). Das Element muss über den Aufruf Add() einem [IComRoutingListEntry](#)-Objekt hinzugefügt werden.

**Rückgabewerte:**

[IComRoutingListItem](#)\*\*.: eine neues [IComRoutingListItem](#) Element

```
HRESULT Add ([in] VARIANT varItem, [in] IComRoutingListItem * pVal)
```

**Beschreibung:** Erstellt ein neues Element vom Typ [IComRoutingListItem](#). Das Element muss über den Aufruf Add() einem [IComRoutingListEntry](#)-Objekt hinzugefügt werden.

**Parameter:**

VARIANT: Bestimmt die Position (0 basiert) des neuen Elements innerhalb [IComRoutingListEntry](#)-Objekts.

[IComRoutingListItem](#)\*\*.: eine neues [IComRoutingListItem](#) Element

HRESULT Remove ([in] VARIANT varItem)

**Beschreibung:** Entfernt ein Element vom Typ [IComRoutingListItem](#) aus dem [IComRoutingListEntry](#)-Objekt.

**Parameter:**

VARIANT: Bestimmt die Position (0 basiert) des zu löschenden Elements innerhalb [IComRoutingListEntry](#)-Objekts .



## IComRoutingListItem Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt ein Element eines Eintrags in der Abarbeitungsliste einer Adhocaktivität.

**Zugriff:** siehe [IComRoutingListEntry](#)

### Öffentliche Methoden

```

HRESULT ID([out, retval] BSTR* pVal)
HRESULT Activity ([in] IComModelActivity* pActivity)
HRESULT ActivityId([out, retval] BSTR* pVal)
HRESULT ActivityId([in] BSTR pVal)
HRESULT ActivityName([out, retval] BSTR* pVal)
HRESULT ActivityName ([in] BSTR newVal)
HRESULT Remark ([out, retval] BSTR* newVal)
HRESULT Remark ([in] BSTR newVal)
HRESULT ReminderPeriodId([out, retval] BSTR* pVal)
HRESULT ReminderPeriodId([in] BSTR Val)
HRESULT ReminderPeriod([in] IComAdhocReminderPeriodTemplate newVal)
HRESULT ReminderPeriodDuration ([out, retval] long* pVal)
HRESULT ReminderPeriodDuration ([in] long newVal)
HRESULT ReminderPeriodIsAbsDuration ([out, retval] long* pVal)
HRESULT ReminderPeriodIsAbsDuration ([in] long newVal)
HRESULT ReminderPeriodTemplates ([out, retval]
IComAdhocReminderPeriodTemplates** pVal)
HRESULT Participants ([out, retval] IComOrgObjectList** pVal)
HRESULT IsDeleteable([out, retval] VARIANT_BOOL* pbVal)
HRESULT IsDeleteable([in] VARIANT_BOOL bVal)
HRESULT IsChangeable([out, retval] VARIANT_BOOL* pbVal)
HRESULT IsChangeable([in] VARIANT_BOOL bVal)

```

### Dokumentation der Elementfunktionen

HRESULT ID ([out, retval] BSTR \* *pVal*)  
**Beschreibung:** Diese Methode liefert die Id des Elements der Abarbeitungsliste.

**Rückgabewerte:**

*BSTR*: Id des Elements der Abarbeitungsliste

HRESULT Activity ([in] [IComModelActivity](#)\* pActivity)  
**Beschreibung:** Diese Methode setzt die Aktivität dieses Elements der Abarbeitungsliste.

**Parameter:**

*IComModelActivity*: Aktivität dieses Elements der Abarbeitungsliste.

HRESULT ActivityId ([out, retval] BSTR \* *pVal*)  
**Beschreibung:** Diese Methode liefert die Id der Aktivität im Modell dieses Elements der Abarbeitungsliste.

**Rückgabewerte:**

*BSTR*: Id der Aktivität im Modell dieses Elements der Abarbeitungsliste.

HRESULT ActivityId ([in] BSTR pVal)

**Beschreibung:** Diese Methode setzt die Id der Aktivität im Modell dieses Elements der Abarbeitungsliste.

**Rückgabewerte:**

*BSTR*: Id der Aktivität im Modell dieses Elements der Abarbeitungsliste.

HRESULT ActivityName ([out, retval] BSTR \* pVal)

**Beschreibung:** Diese Methode liefert den Namen der Aktivität dieses Elements der Abarbeitungsliste.

**Rückgabewerte:**

*BSTR*: Name der Aktivität dieses Elements der Abarbeitungsliste.

HRESULT ActivityName ([in] BSTR newVal)

**Beschreibung:** Diese Methode setzt den Namen der Aktivität dieses Elements der Abarbeitungsliste.

**Parameter:**

*BSTR*: neuer Aktivitätsname.

HRESULT Remark ([out, retval] BSTR\* newVal)

**Beschreibung:** Diese Methode liefert die Bemerkung zu diesem Element der Abarbeitungsliste.

**Rückgabewerte:**

*BSTR*: Bemerkung zu diesem Element der Abarbeitungsliste.

HRESULT Remark ([in] BSTR newVal)

**Beschreibung:** Diese Methode setzt die Bemerkung zu diesem Element der Abarbeitungsliste.

**Parameter:**

*BSTR*: neue Bemerkung.

HRESULT ReminderPeriodId ([out, retval] BSTR \* pVal)

**Beschreibung:** Diese Methode liefert die Id der Mahnfristenvorlage für dieses Element der Abarbeitungsliste.

**Rückgabewerte:**

*BSTR*: Id der Mahnfristenvorlage für dieses Element der Abarbeitungsliste.

HRESULT ReminderPeriod ([in] [IComAdhocReminderPeriodTemplate](#) period)

**Beschreibung:** Diese Methode setzt die Mahnfristenvorlage für dieses Element der Abarbeitungsliste. Intern wird auch der Vorlage entsprechend der Name und die Dauer der Frist gesetzt.

**Parameter:**

*BSTR*: die Mahnfristenvorlage für dieses Element der Abarbeitungsliste

HRESULT ReminderPeriodDuration ([out, retval] long \* pVal)

**Beschreibung:** Diese Methode liefert die Dauer der eingerichteten Frist in Sekunden. Je nach dem wie ReminderPeriodIsAbsDuration eingerichtet ist, ist dies ein absoluter oder relativer Wert.

**Rückgabewerte:**

*long*: die Dauer der eingerichteten Frist

HRESULT ReminderPeriodDuration([in] long newVal)

**Beschreibung:** Diese Methode setzt die Dauer der eingerichteten Frist in Sekunden. Je nach dem wie ReminderPeriodIsAbsDuration eingerichtet ist, ist dies ein absoluter oder relativer Wert.

**Parameter:**

*long*: Dauer der eingerichteten Frist in Sekunden

HRESULT ReminderPeriodIsAbsDuration ([out, retval] VARIANT\_BOOL \* pVal)

**Beschreibung:** Diese Methode gibt an, ob die konfigurierte Dauer der Frist ein absoluter oder ein relativer Wert ist.

**Rückgabewerte:**

*VARIANT\_BOOL*: true, wenn die Dauer der Frist ein absoluter Wert ist; sonst false.

HRESULT ReminderPeriodIsAbsDuration([in] VARIANT\_BOOL newVal)

**Beschreibung:** Diese Methode setzt, ob die konfigurierte Dauer der Frist ein absoluter oder ein relativer Wert ist.

**Parameter:**

*VARIANT\_BOOL*: true, wenn die Dauer der Frist ein absoluter Wert ist; sonst false.

HRESULT ReminderPeriodTemplates ([out, retval]

[IComAdhocReminderPeriodTemplates](#)\*\* pVal)

**Beschreibung:** Diese Methode liefert die Vorlagen für mögliche Fristen des RoutingListItems.

**Rückgabewerte:**

[IComAdhocReminderPeriodTemplates](#)\*\*: Vorlagen für mögliche Fristen des RoutingListItems.

HRESULT Participants ([out, retval] [IComOrgObjectList](#)\*\* pVal)

**Beschreibung:** Diese Methode liefert die Teilnehmer des RoutingListItems als veränderbare Liste von [IComOrgObjectList](#).

**Rückgabewerte:**

[IComOrgObjectList](#)\*\*: die Teilnehmer des RoutingListItems als veränderbare Liste von IOrgObject.

HRESULT IsDeleteable([out, retval] VARIANT\_BOOL\* pbVal)

**Beschreibung:** Diese Methode gibt an, ob das RoutingListItem löscher ist.

**Rückgabewerte:**

*VARIANT\_BOOL*: true, wenn das RoutingListItem löscher ist.

HRESULT IsDeleteable([in] VARIANT\_BOOL bVal)

**Beschreibung:** Diese Methode setzt, ob das RoutingListItem löscher ist.

**Parameter:**

*VARIANT\_BOOL*: true, wenn das RoutingListItem löscher ist

HRESULT IsChangeable([out, retval] VARIANT\_BOOL\* pbVal)

**Beschreibung:** Diese Methode gibt an, ob das RoutingListItem änderbar ist.

**Rückgabewerte:**

VARIANT\_BOOL: true, wenn das RoutingListItem änderbar ist.

HRESULT IsChangeable([in] VARIANT\_BOOL bVal)

**Beschreibung:** Diese Methode setzt, ob das RoutingListItem änderbar ist.

**Parameter:**

*VARIANT\_BOOL*: true, wenn das RoutingListItem änderbar ist

## IComAdhocReminderPeriodTemplates Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt die konfigurierten Mahnfristen einer Ad-hoc-aktivität. Diese Schnittstelle ist von [IComCollection](#) abgeleitet und enthält Elemente vom Typ [IComAdhocReminderTemplate](#).

**Zugriff auf die Liste:** siehe [IComCollection](#)

**Elemente der Liste:** siehe [IComAdhocReminderTemplate](#)

## IComAdhocReminderPeriodTemplate Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle ermöglicht den Zugriff auf eine vorkonfigurierte Mahnfrist einer Ad-hoc-aktivität.

### Öffentliche Methoden

HRESULT [Id](#)([out, retval] BSTR\* pVal)  
HRESULT [Name](#)([out, retval] BSTR\* pVal)  
HRESULT [DefaultDuration](#) ([out,retval] long\* lVal)

### Dokumentation der Elementfunktionen

HRESULT [Id](#)([out, retval] BSTR\* pVal)

**Beschreibung:** Die Methode liefert die Id der Mahnfrist.

**Rückgabewerte:**

[IComRoutingListEntry](#) BSTR: Id

HRESULT [Name](#)([out, retval] [IComRoutingListEntry](#)\*\* pEntry)

**Beschreibung:** Diese Methode liefert den Namen der Mahnfrist.

**Rückgabewerte:**

BSTR: Name der Mahnfrist

HRESULT [DefaultDuration](#)([out, retval] long\* pEntry)

**Beschreibung:** Diese Methode liefert die vorkonfigurierte Dauer der Mahnfrist in Sekunden.

**Rückgabewerte:**

long\*: Dauer in Sekunden

## IScriptResult Schnittstellenreferenz

### Ausführliche Beschreibung

Mit Hilfe dieser Schnittstelle kann das Programmverhalten nach Ausführung eines VB-Skripts bestimmt werden. Die Schnittstelle ist in den Client-Events unter dem Namen 'ScriptResult' verfügbar.

### Öffentliche Methoden

```
HRESULT ResultCode ([out, retval] short *pResultCode)
HRESULT ResultCode ([in] short nResultCode)
HRESULT ResultString ([out, retval] BSTR *pVal)
HRESULT ResultString ([in] BSTR newVal)
HRESULT ShowResultString ([out, retval] VARIANT_BOOL *pVal)
HRESULT ShowResultString ([in] VARIANT_BOOL newVal)
```

### Dokumentation der Elementfunktionen

HRESULT ResultCode ([out, retval] short \* *pResultCode*)

**Beschreibung:** Diese Methode liefert den Ergebnisstatus des ausgeführten Skripts. Die Methode sollte im Event BEFORE\_FORWARD ausgewertet werden.

**Rückgabewerte:**

*short:* Status des Skripts

**Hinweis:** mögliche Statuswerte

- § 0 = Vorgangsschritt kann weitergeleitet werden
- § -1 = Abbruch des Weiterleitens, Maske bleibt geöffnet, Daten werden nicht gespeichert
- § -2 = Abbruch des Weiterleitens, Maske wird geschlossen, Daten werden gespeichert
- § -3 = Abbruch des Weiterleitens, Maske wird geschlossen, Daten werden nicht gespeichert

HRESULT ResultCode ([in] short *nResultCode*)

**Beschreibung:** Mit dieser Methode kann das Weiterleiten eines Vorganges unterbunden werden. Die Methode sollte im Event BEFORE\_FORWARD benutzt werden.

**Parameter:**

*short*: Status

**Hinweis:** mögliche Statuswerte

- § 0 = Vorgangsschritt kann weitergeleitet werden
- § -1 = Abbruch des Weiterleitens, Maske bleibt geöffnet, Daten werden nicht gespeichert
- § -2 = Abbruch des Weiterleitens, Maske wird geschlossen, Daten werden gespeichert
- § -3 = Abbruch des Weiterleitens, Maske wird geschlossen, Daten werden nicht gespeichert

**Beispiel:** Ein Maskenfeld soll bestimmten Wert (MyValue) haben. Wenn nicht, kommt eine Fehlermeldung und der Benutzer muss die Eingabe nochmals machen.

```
Dim MyValue

IF wfvariables.MyVariable.value < MyValue Then 'MyVariable -> Name der
                                                'Variable die im WF-
Editor                                          'dem Maskenfeld
                                             zugeordnet
                                                'wurde

    'Fehlermeldung setzen
    scriptresult.ResultString = "Variable X muss mindestens den Wert Y
    haben!"

    'Meldung soll angezeigt werden
    scriptresult.ShowResultString = True

    'Abbruch des Weiterleitens, Maske bleibt geöffnet, Daten werden nicht
    'gespeichert
    scriptresult.ResultCode = -1
End if
```

HRESULT ResultString ([out, retval] BSTR \* *pVal*)

**Beschreibung:** Diese Methode liefert den Ergebnistext oder die Fehlermeldung, wenn die Eigenschaft ShowResultString auf true gesetzt ist.

**Rückgabewerte:**

*BSTR*: Ergebnistext oder Fehlermeldung

**Siehe auch:**

[ShowResultString](#)

HRESULT ResultString ([in] BSTR *newVal*)

**Beschreibung:** Diese Methode liefert den Ergebnistext oder die Fehlermeldung, wenn die Eigenschaft ShowResultString auf TRUE gesetzt ist.

**Rückgabewerte:**

*BSTR*: Ergebnistext oder Fehlermeldung

**Beispiel:** siehe [ResultCode](#)

HRESULT ShowResultString ([out, retval] VARIANT\_BOOL \* *pVal*)

**Beschreibung:** Diese Methode gibt an, ob der Ergebnistext bzw. die Fehlermeldung angezeigt wird.

**Rückgabewerte:**

*VARIANT\_BOOL*: True = Text wird angezeigt

HRESULT ShowResultString ([in] VARIANT\_BOOL *newVal*)

**Beschreibung:** Diese Methode bestimmt, ob der Ergebnistext angezeigt werden soll.

**Parameter:**

*VARIANT\_BOOL*: True = Text wird angezeigt

**Beispiel:** siehe [ResultCode](#)



## IWFClient Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle dient zur Verwaltung der Zugriffe auf das Workflowsystem. Sie ermöglicht den Zugriff auf den Eingangskorb, sowie die startbaren Vorgänge, auf einzelne Vorgänge und dessen Variablen usw.

### Öffentliche Methoden

HRESULT [SetUserAbsence](#) ([in] VARIANT\_BOOL bAbsent,[out]VARIANT \*pvarErrorString,[out, retval]long \*lpResult)  
HRESULT [CurrentUserAbsent](#) ([out, retval] VARIANT\_BOOL \*pVal)  
HRESULT [GetActivityPerformer](#) ([in]BSTR bstrActivityID,[out, retval]BSTR \*pbstrPerformers)  
HRESULT [UserName](#) ([out, retval] BSTR \*pVal)

### Dokumentation der Elementfunktionen

HRESULT SetUserAbsence ([in] VARIANT\_BOOL *bAbsent*, [out] VARIANT \**pvarErrorString*, [out, retval] long \* *lpResult*)

**Beschreibung:** Diese Methode bestimmt die Anwesenheit des aktuell angemeldeten Benutzers.

**Parameter:**

*VARIANT\_BOOL*: TRUE, wenn der Benutzer abgemeldet werden soll, ansonsten FALSE.  
*VARIANT*: hier wird ggf. der Fehlertext zurückgeliefert

**Rückgabewerte:**

*long*: 0 = kein Fehler aufgetreten

**Beispiel:**

```
Result = wfclient.SetUserAbsence(TRUE, ErrorString)
```

HRESULT CurrentUserAbsent ([out, retval] VARIANT\_BOOL \* *pVal*)

**Beschreibung:** Diese Methode gibt an, ob der aktuelle Benutzer abwesend gemeldet ist.

**Rückgabewerte:**

*VARIANT\_BOOL*: TRUE = Benutzer abwesend

**Beispiel:**

```
isabsend = wfclient.CurrentUserAbsent
```

HRESULT GetActivityPerformer ([in] BSTR *bstrRActivityID*, [out, retval] BSTR \* *pbstrPerformers*)

**Beschreibung:** Diese Methode liefert die Id des Benutzer zu einer bestimmten Aktivität.

**Parameter:**

*BSTR*: InstanzId der Aktivität

**Rückgabewerte:**

*BSTR*: ID des Bearbeiters

**Beispiel:**

```
Dim MyInstanceId 'Instanzid der Aktivität  
AdvisorId = wfclient.GetActivityPerformers(MyInstanceId)
```

HRESULT UserName ([out, retval] BSTR \* *pVal*)

**Beschreibung:** Diese Methode liefert den Namen des aktuell angemeldeten Benutzers.

**Rückgabewerte:**

*BSTR*: Benutzername

**Beispiel:**

```
UserName = wfclient.UserName
```

## IWFFieldValues Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine zu einem Textfeld hinterlegte Auswahlliste. Die Schnittstelle ist aufgebaut wie eine '.ini' Datei im Speicher. Zugriff auf diese Schnittstelle in Client-Events:

```
Dim MaskfieldName 'Name des Textfeldes

Set Maskfield = wfmaskfields.item(MaskfieldName)

Set MaskfieldListe = Maskfield.WFFieldValues
```

### Öffentliche Methoden

HRESULT [WriteStringValue](#) ([in] BSTR strSection,[in] BSTR strEntry,[in] BSTR strValue)

HRESULT [GetStringValue](#) ([in] BSTR strSection,[in] BSTR strEntry,[in] BSTR strDefault,[out, retval] BSTR \*strValue)

## Dokumentation der Elementfunktionen

HRESULT WriteStringValue ([in] BSTR *strSection*, [in] BSTR *strEntry*, [in] BSTR *strValue*)

**Beschreibung:** Diese Methode fügt einen Wert in eine Liste, die einem Textfeld zugeordnet ist, ein. Der Wert kann immer nur ans Ende der Liste angefügt werden. Ein Überschreiben bestehender Listenelemente ist nicht möglich.

**Parameter:**

*BSTR*: Sektionsname

*BSTR*: Listenposition z.B. Zeile3

*BSTR*: Neuer Wert

**Hinweis:** Sektionsname siehe [IWFMaskField::SectionName](#)

**Beispiel:**

```
Dim MaskfieldName 'Name des Textfeldes
Dim MySectionName 'Sektionsname
Dim MyPropertyName 'Zeilenummer
Dim MyDefaultValue 'Vorbelegung
Dim MyNewValue 'Neuer Wert

MyDefaultValue = "free entry"

Set Maskfield = wfmaskfields.item(MaskfieldName)

MySectionName = Maskfield.SectionName

Set MaskfieldListe = Maskfield.WFFieldValues

count = 0
Do
    MyPropertyName = "Zeile" & count

    Value = MaskfieldListe.GetStringValue(MySectionName, _
        MyPropertyName, MyDefaultValue)

    If Value = "free entry" Then
        MaskfieldListe.WriteStringValue MySectionName, MyPropertyName, _
            MyNewValue
        Exit Do
    End If
    count = count + 1
Loop
```

HRESULT GetStringValue ([in] BSTR *strSection*, [in] BSTR *strEntry*, [in] BSTR *strDefault*, [out, retval] BSTR \* *strValue*)

**Beschreibung:** Diese Methode liefert den Wert aus einer Liste, die einem Textfeld zugeordnet ist.

**Parameter:**

*BSTR*: Sektionsname

*BSTR*: Listenposition z.B. Zeile3

*BSTR*: Default, dieser Wert wird zurückgeliefert wenn kein Wert gefunden wurde

**Rückgabewerte:**

*BSTR*: Wert aus der Liste

**Beispiel:** siehe [WriteStringValue](#)

## IWFFile Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt die Workflowakte eines Vorgangsschrittes. Die Schnittstelle ist in den Client-Events unter dem Namen 'wffile' verfügbar.

### Öffentliche Methoden

```
HRESULT FileObjects ([out, retval] IFileObjects **pVal)
HRESULT AddFileObject ([in] long lObjectID,[in] long lObjectType,[in]
VARIANT_BOOL bInWorkspace,[in] DocLocationEnum location,[out, retval]
VARIANT_BOOL *pbResult)
HRESULT DeleteFileObject ([in] long lObjectID,[out, retval] VARIANT_BOOL
*pbResult)
HRESULT GetFileObjectByID ([in] long lObjectID,[out, retval] IFileObject
**ppFileObject)
```

### Dokumentation der Elementfunktionen

```
HRESULT FileObjects ([out, retval] IFileObjects ** pVal)
```

**Beschreibung:** Diese Methode liefert eine Liste aller Dokumente, die in der Workflowakte enthalten sind.

**Rückgabewerte:**

*IFileObjects*: Liste aller Dokumente

**Beispiel:**

```
Set DocumentsList = wffile.FileObjects
```

HRESULT AddFileObject ([in] long *lObjectID*, [in] long *lObjectType*, [in] VARIANT\_BOOL *bInWorkspace*, [in] DocLocationEnum *location*, [out, retval] VARIANT\_BOOL \* *pbResult*)

**Beschreibung:** Diese Methode fügt der Workflowakte ein Dokument hinzu.

**Parameter:**

*long*: ID des neuen Objekts

*long*: Objekttyp

*VARIANT\_BOOL*: gibt an, ob das Objekt in den Arbeitsbereich der Akte gehört

*DocLocationEnum*: 1 = Objekt bereits zugeordnet 2 = Objekt liegt in der WF-Ablage

**Rückgabewerte:**

*VARIANT\_BOOL*: TRUE = Objekt hinzugefügt, andernfalls FALSE

**Beispiel:** Das Dokument wird in den Arbeitsbereich der WF-Akte eingefügt.

```
Dim MyDocId      'Dokumentenid aus dem DMS
Dim MyDocType    'Dokumententyp aus dem DMS

added = wffile.AddFileObject(MyDocId, MyDocType, TRUE, 1)
```

HRESULT DeleteFileObject ([in] long *lObjectID*, [out, retval] VARIANT\_BOOL \* *pbResult*)

**Beschreibung:** Diese Methode löscht ein Dokument aus der Akte.

**Parameter:**

*long*: ID des Objekts

**Rückgabewerte:**

*VARIANT\_BOOL*: TRUE = Objekt entfernt, andernfalls FALSE

**Beispiel:**

```
Dim MyDocId      'Dokumentenid aus dem DMS

deleted = wffile.DeleteFileObject(MyDocId)
```

HRESULT GetFileObjectByID ([in] long *lObjectID*, [out, retval] [IFileObject](#) \*\*  
*ppFileObject*)

**Beschreibung:** Diese Methode liefert ein Dokumentenobjekt aus der Workflowakte, anhand der gegebenen Dokumentenid aus dem **enaio**®.

**Parameter:**

*long*: Objekt-ID

**Rückgabewerte:**

*IFileObject*: Dokumentenobjekt

**Beispiel:**

```
Dim MyDocId      'Dokumentenid aus dem DMS

Set Document = wffile.GetFileObjectByID(MyDocId)
```



## IWFListColumn Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Spalte eines Tabellenfeldes in Eingabe-Masken.

### Öffentliche Methoden

HRESULT [Name](#) ([out, retval] BSTR \*pVal)  
HRESULT [DataType](#) ([out, retval] BSTR \*pVal)  
HRESULT [GetStringProperty](#) ([in] BSTR strPropertyName,[in] BSTR strDefault,[out, retval] BSTR \*pbstrProperty)  
HRESULT [SetStringProperty](#) ([in]BSTR strPropertyName,[in] BSTR strValue,[out, retval] VARIANT\_BOOL \*pbResult)

### Dokumentation der Elementfunktionen

HRESULT Name ([out, retval] BSTR \* *pVal*)  
**Beschreibung:** Diese Methode liefert den Spaltennamen.

**Rückgabewerte:**

*BSTR*: Spaltenname

HRESULT DataType ([out, retval] BSTR \* *pVal*)  
**Beschreibung:** Diese Methode liefert den Spaltentyp.

**Rückgabewerte:**

*BSTR*: Spaltentyp

HRESULT GetStringProperty ([in] BSTR *strPropertyName*, [in] BSTR *strDefault*, [out, retval] BSTR \* *pbstrProperty*)

**Beschreibung:** Diese Methode liefert einen Eigenschaftswert einer Spalte. So können auch Listeneinträge einer Kombobox für die Spalte abgefragt werden.

**Parameter:**

*BSTR*: Name der Eigenschaft, z.B. ZEILE1

*BSTR*: dieser Wert wird zurückgegeben, wenn die abgefragte Eigenschaft nicht vorhanden ist.

**Rückgabewerte:**

*BSTR*: der Wert der Eigenschaft

**Beispiel:** Ein Wert einer Kombobox wird ermittelt.

```
Dim MaskfieldName    'Name des Maskenfelds in Tabellenform
Dim MyPosition        'Position der Spalten
Dim MyPropertyName    'Name der Eigenschaft, z.B. ZEILE2
Dim MyDefaultValue    'Falls Eigenschaft nicht vorhanden, dann wird
                      'dieser Wert zurückgegeben

Set ListColumns = wfmaskfields.item(MaskfieldName).WFListColumns

Set Column = ListColumns.item(MyPosition)

Value = Column.GetStringProperty(PropertyName, MyDefaultValue)
```

**Siehe auch:**

[GetStringProperty\(\)](#)

HRESULT SetStringProperty ([in] BSTR *strPropertyName*, [in] BSTR *strValue*, [out, retval] VARIANT\_BOOL \* *pbResult*)

**Beschreibung:** Diese Methode setzt einen Eigenschaftswert. Sind Einträge in der Form ZEILE1 bis ZEILEn hinterlegt, werden diese in Form einer Kombinationsbox in der Spalte dargestellt.

**Parameter:**

*strPropertyName* Name der Eigenschaft

*strValue* Wert der Eigenschaft

**Rückgabewerte:**

\**pbResult* TRUE, kein Fehler, andernfall FALSE

**Beispiel:** Ein Wert (My\_New\_Value) wird in Kombobox eingefügt.

```
Dim MaskfieldName      'Name des Maskenfelds in Tabellenform
Dim MyPosition         'Position der Spalten
Dim MyPropertyName     'Name der Eigenschaft
Dim MyDefaultValue     'Falls Eigenschaft nicht vorhanden, dann wird dieser
                        'Wert zurückgegeben z.B. free entry
Dim MyNewValue         'Wert, der in die Kombobox aufgenommen werden soll

MyDefaultValue = "free entry"

Set Maskfield = wfmaskfields.item(MaskfieldName) 'Objekt des
Maskenfeldes

Set Column = Maskfield.WFListColumns.item(MyPosition) 'Spaltenobjekt

count = 0
Do 'ersten freien Listeneintrag suchen
    PropertyName = "Zeile" & count 'Eigenschaftsname zusammensetzen
    'Listeneinträge holen
    Value = Column.GetStringProperty(PropertyName, MyDefaultValue)

    If Value = "free entry" Then 'freien Listeneintrag gefunden
        'neuer Eintrag
        Result = Column.SetStringProperty(PropertyName, MyNewValue)
        Exit Do
    End If
    count = count + 1
Loop
```

**Siehe auch:**

[GetStringProperty\(\)](#)

## IWFListColumns Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Liste aller Spalten eines Tabellenfeldes in Eingabe-Masken. Die Schnittstelle ist von [IComCollection](#) abgeleitet. Die Liste wird über folgenden Aufruf erstellt:

```
Dim MaskfieldName  `Name des Maskenfelds

`das Maskenfeld muss eine Tabelle sein
Set Maskfield = wfmaskfields.item(MaskfieldName)

Set ColumnList = Maskfield.WFListColumns
```

**Elemente der Liste:** siehe [IWFListColumn](#)

### Öffentliche Methoden

HRESULT [TableName](#) ([out, retval] BSTR \*pVal)

### Dokumentation der Elementfunktionen

HRESULT TableName ([out, retval] BSTR \* *pVal*)

**Beschreibung:** Diese Methode liefert Namen der Datenbanktabelle für diese Tabelle.

**Rückgabewerte:**

*BSTR*: Name der Datenbanktabelle

## IWFMaskField Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt ein Maskenfeld in Eingabe-Masken. In Client-Event erhält man ein Objekt folgendermaßen:

```
Dim MaskfieldName 'Name des Maskenfelds

Set Maskfield = wfmaskfields.item(MaskfieldName)
```

### Öffentliche Methoden

```
HRESULT CtrlPages ([out, retval] IWFCtrlPages* *pVal)
HRESULT WFFieldValues ([out, retval] IWFFieldValues **ppValues)
HRESULT Name ([out, retval] BSTR *pVal)
HRESULT DataType ([out, retval] BSTR *pVal)
HRESULT WFListColumns ([out, retval] IWFListColumns **ppListColumns)
HRESULT SectionName ([out, retval] BSTR *pVal)
HRESULT Visible ([out, retval] VARIANT_BOOL *pVal)
HRESULT Visible ([in] VARIANT_BOOL newVal)
HRESULT Enabled ([out, retval] VARIANT_BOOL *pVal)
HRESULT Enabled ([in] VARIANT_BOOL newVal)
HRESULT ID ([out, retval] BSTR *pVal)
```

### Dokumentation der Elementfunktionen

```
HRESULT WFFieldValues ([out, retval] IWFFieldValues ** ppValues)
```

**Beschreibung:** Diese Methode erlaubt den Zugriff auf die Liste, die zu einem Textfeld hinterlegt wurde.

**Rückgabewerte:**

*IWFFieldValues*: Liste zum Textfeld

**Beispiel:**

```
Dim MaskfieldName 'Name des Maskentextfelds

Set ListField = wfmaskfields.item(MaskfieldName).WFFieldValues
```

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

**Beschreibung:** Diese Methode liefert den Name des Maskenfeldes.

**Rückgabewerte:**

*BSTR*: Name des Maskenfeldes

**Beispiel:**

```
Dim Index    'Position des Maskenfelds in Liste

MaskfieldName = wfmaskfields.item(Index).Name
```

HRESULT DataType ([out, retval] BSTR \* *pVal*)

**Beschreibung:** Diese Methode liefert den Datentyp des Maskenfeldes.

**Rückgabewerte:**

*BSTR*: Datentyp des Maskenfeldes

**Hinweis:** mögliche Werte

Ohne Einschränkung

- § X = alle Zeichen
- § A = Buchstaben
- § D = Datum (Länge 10)
- § # = Dezimalzahlen (Länge 15)
- § 9 = Ziffern (Länge 9)
- § Z = Ziffern(alphanumerisch)
- § I = Archivlistenfeld (nur für Mehrfachfelder)

Mit Einschränkung

- § P = Patientenart (a/s/f)
- § T = Seite (l/r)
- § S = Geschlecht (m/w)
- § Q = Frage (j/n)
- § F = nur Großbuchstaben (beim Schreiben von dataType)
- § G = nur Großbuchstaben (beim Lesen von dataType)
- §

**Beispiel:**

```
Dim MaskfieldName    'Name des Maskenfelds

DataType = wfmaskfields.item(MaskfieldName).DataType
```

HRESULT WFListColumns ([out, retval] [IWFListColumns](#) \*\* ppListColumns)

**Beschreibung:** Diese Methode liefert eine Liste aller Tabellenspalten. Diese Methode ist nur für Maskenfelder in Tabellenform gültig.

**Rückgabewerte:**

*IWFListColumns*: Liste aller Tabellenspalten

**Beispiel:**

```
Dim MaskfieldName `Name des Maskenfelds in Tabellenform

Set ListColumns = wfmaskfields.item(MaskfieldName).WFListColumns
```

HRESULT SectionName ([out, retval] BSTR \* pVal)

**Beschreibung:** Diese Methode liefert den Sektionsnamen (Name des Vorgangsschritts und Feldname). Die beiden Strings sind durch das Pipezeichen "|" getrennt. Der Sektionsname wird für den Zugriff auf eine Liste an Textfeldern benötigt.

**Rückgabewerte:**

*BSTR*: Vorgangsschritt|Maskenfeld

**Beispiel:**

```
Dim MaskfieldName `Name des Maskenfelds in Tabellenform

SectionName = wfmaskfields.item(MaskfieldName).SectionName
```

**Siehe auch:**

[IWFFieldValues](#)

HRESULT Visible ([out, retval] VARIANT\_BOOL \* pVal)

**Beschreibung:** Diese Methode gibt an, ob das Maskenfeld sichtbar ist.

**Rückgabewerte:**

*VARIANT\_BOOL*: TRUE = Feld sichtbar

**Beispiel:**

```
Dim MaskfieldName `Name des Maskenfelds

isvisible = wfmaskfields.item(MaskfieldName).Visible
```

HRESULT Visible ([in] VARIANT\_BOOL *newVal*)

**Beschreibung:** Mit Hilfe dieser Methode kann man ein Maskenfeld sichtbar/unsichtbar machen.

**Parameter:**

*VARIANT\_BOOL*: TRUE = Feld sichtbar, ansonsten FALSE

**Beispiel:** Das Maskenfeld ist für den Benutzer nicht mehr sichtbar.

```
Dim MaskfieldName    'Name des Maskenfelds

wfmaskfields.item(MaskfieldName).Visible = False
```

HRESULT Enabled ([out, retval] VARIANT\_BOOL \* *pVal*)

**Beschreibung:** Diese Methode gibt an, ob das Maskenfeld für die Eingabe frei/gesperrt ist.

**Rückgabewerte:**

*VARIANT\_BOOL*: TRUE = frei, FALSE = Feld gesperrt

**Beispiel:**

```
Dim MaskfieldName    'Name des Maskenfelds

isEnabled = wfmaskfields.item(MaskfieldName).Enabled
```

HRESULT Enabled ([in] VARIANT\_BOOL *newVal*)

**Beschreibung:** Mit Hilfe dieser Methode kann man ein Maskenfeld für die Eingabe sperren.

**Parameter:**

*VARIANT\_BOOL*: FALSE = Feld gesperrt

**Beispiel:** Das Maskenfeld ist für Benutzereingaben gesperrt.

```
Dim MaskfieldName    'Name des Maskenfelds

wfmaskfields.item(MaskfieldName).Enabled = False
```



HRESULT ID ([out, retval] BSTR \* *pVal*)

**Beschreibung:** Diese Methode liefert die ID der Maskenfelds.

**Rückgabewerte:**

*BSTR*: MaskenfeldId

**Beispiel:** Das Maskenfeld ist für Benutzereingaben gesperrt.

```
Dim MaskfieldName  'Name des Maskenfelds

MaskfieldId = wfmaskfields.item(MaskfieldName).ID
```

HRESULT CtrlPages ([out, retval] [IWFCtrlPages](#)\* \*pVal)

**Beschreibung:** Diese Methode liefert eine Liste aller Seiten des Pagecontrols. Die Methode ist nur bei Maskenfeldern vom DataType = C gültig:

**Rückgabewerte:**

*IWFCtrlPages*: Liste aller Seiten des Pagecontrols

**Beispiel:** Das Maskenfeld ist für Benutzereingaben gesperrt.

```
Dim PageCtrlName  'Name des PageControls

Set oPages = wfmaskfields.item(PageCtrlName).CtrlPages
```

## IWFMaskFields Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Liste aller Maskenfelder. Diese Schnittstelle ist von [IComCollection](#) abgeleitet. Die Liste steht in Client-Events unter dem Namen 'wfmaskfields' zur Verfügung.

**Elemente der Liste:** siehe [IWFMaskField](#)

## IWFCtrlPage Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Seite eines Pagecontrols. In Client-Event erhält man ein Objekt folgendermaßen:

```
Set oPage = wfmaskfields.item([Name PageControl]).ctrlpages.item([Seitenname o.  
Index])
```

### Öffentliche Methoden

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

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

HRESULT [PageNumber](#)([out, retval] long\* pVal)

### Dokumentation der Elementfunktionen

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

**Beschreibung:** Diese Methode liefert eine Liste aller Maskenfelder, die sich auf dieser Seite befinden.

**Rückgabewerte:**

*IWFMaskFields*: Liste aller Felder

**Beispiel:**

```
Set oFields = WFMaskFields("PageCtrl").CtrlPages.Item(0).Fields
```

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

**Beschreibung:** Diese Methode liefert den Namen der Seite.

**Rückgabewerte:**

*BSTR*: Name der Seite

**Beispiel:**

```
PageName = WFMaskFields("PageCtrl").CtrlPages.Item(0).Name
```

HRESULT PageNumber([out, retval] long\* pVal)

**Beschreibung:** Diese Methode liefert den Namen der Seite.

**Rückgabewerte:**

*long*: Seitennummer

**Beispiel:**

```
Number = WFMaskFields("PageCtrl4").CtrlPages.Item("Seite1").PageNumber
```

## IWFCtrlPages Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Liste aller Seiten eines Pagecontrols auf der Maske. Diese Schnittstelle ist von [IComCollection](#) abgeleitet.

**Elemente der Liste:** siehe [IWFCtrlPage](#)

### Öffentliche Methoden

HRESULT [ActivePage](#)([out, retval] short \*pVal)

HRESULT [ActivePage](#)([in] short nPage)

HRESULT [EnablePage](#)([in] short nPage, [in] VARIANT\_BOOL bEnable)

HRESULT [PageEnabled](#)([in] short nPage, [out, retval] VARIANT\_BOOL\* pbEnabled)

### Dokumentation der Elementfunktionen

HRESULT ActivePage([out, retval] short \*pVal)

**Beschreibung:** Diese Methode liefert die Nummer der aktuell aktivierten Seite.

**Rückgabewerte:**

*short*: Nummer der aktivierten Seite (nullbasiert)

**Beispiel:**

```
PageNumber = WFMaskFields("PageCtrl4").CtrlPages.ActivePage
```

HRESULT ActivePage([in] short nPage)

**Beschreibung:** Diese Methode aktiviert die angebene Seite.

**Parameter:**

short: Nummer der Seite, die aktiviert werden soll (nullbasiert)

**Beispiel:**

```
WfMaskFields("PageCtrl4").CtrlPages.ActivePage = 2
```

HRESULT EnablePage([in] short nPage, [in] VARIANT\_BOOL bEnable)

**Beschreibung:** Diese Methode schaltet den Schreibschutz einer Seite ein/aus.

**Parameter:**

short: Nummer der Seite (nullbasiert)

VARIANT\_BOOL: True = Schreibschutz deaktiviert, False = Schreibschutz aktiviert

**Beispiel:** Aktivierung des Schreibschutzes der 2. Seite eines Pagecontrols.

```
Dim CtrlPageName 'Name des Pagecontrol

WfMaskFields(CtrlPageName).CtrlPages.EnablePage 1, False
```

HRESULT PageEnabled([in] short nPage, [out, retval] VARIANT\_BOOL\* pbEnabled)

**Beschreibung:** Diese Methode schaltet gibt an, ob sich auf der angegebenen Seite ein Schreibschutz befindet.

**Parameter:**

short: Nummer der Seite (nullbasiert)

**Rückgabewerte:**

VARIANT\_BOOL: True = Schreibschutz deaktiviert, False = Schreibschutz aktiviert

**Beispiel:** Ist die erste Seite aktiv?

```
Dim CtrlPageName 'Name des Pagecontrol

enabled = wfmaskfields.item(CtrlPageName).CtrlPages.PageEnabled(0)
```

## IWFVariable Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Workflowvariable. Da Variablen verschachtelt sein können, besitzt dieses Objekt die Eigenschaften und Methoden einer Liste. In Client-Events spricht man ein Objekt dieser Schnittstelle folgendermaßen an:

```

wfvariables.MyWorkflowvariable  'MyWorkflowvariable -> Hier steht der
                                'im Workflow-Editor definierte
                                'Variablenname

```

Siehe auch: [IWFVariables](#)

## Öffentliche Methoden

HRESULT [Name](#) ([out, retval] BSTR \*pVal)  
 HRESULT [ID](#) ([out, retval] BSTR \*pVal)  
 HRESULT [Value](#) ([out, retval] BSTR \*pVal)  
 HRESULT [Value](#) ([in] BSTR newVal)  
 HRESULT [VarType](#) ([out, retval] VarTypeEnum \*pVal)  
 HRESULT [VarDataType](#) ([out, retval] VarDataTypeEnum \*pVal)  
 HRESULT [FieldName](#) ([out, retval] BSTR \*pVal)  
 HRESULT [InputMode](#) ([out, retval] long \*pVal)  
 HRESULT [InitialValue](#) ([out, retval] BSTR \*pVal)

## Listenmethoden

Diese Methoden können nur bei Workflowvariablen in Listenform benutzt werden.

HRESULT [Item](#) ([in] VARIANT varItem, [out, retval] [IWFVariable](#) \*\*pVal)  
 HRESULT [Count](#) ([out, retval] long \*pVal)  
 HRESULT [CreateListItem](#) ([out, retval] [IWFVariable](#) \*\*ppVariable)  
 HRESULT [AddListItem](#) ([in] [IWFVariable](#) \*pListItem)  
 HRESULT [RemoveListItem](#) ([in] VARIANT varItem)  
 HRESULT [ListType](#) ([out, retval] ListTypeEnum \*pVal)  
 HRESULT [Selected](#) ([out, retval] VARIANT\_BOOL \*pVal)  
 HRESULT [Selected](#) ([in] VARIANT\_BOOL newVal)  
 HRESULT [TypeId](#) ([out, retval] BSTR \*pVal)  
 HRESULT [ClearList](#)()

## Dokumentation der Elementfunktionen

HRESULT Item ([in] VARIANT *varItem*, [out, retval] [IWFVariable](#) \*\* *pVal*)

**Beschreibung:** Diese Methode liefert ein Listenelement.

**Rückgabewerte:**

*IWFVariable*: Listenelement

**Beispiel:**

```
'erste Element einer Liste  
Set Element = wfvariables.MyList.item(0)
```

HRESULT Count ([out, retval] long \* *pVal*)

**Beschreibung:** Diese Methode liefert die Anzahl der Listenelemente.

**Rückgabewerte:**

*long*: Anzahl

**Beispiel:**

```
count = wfvariables.MyList.count
```

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

**Beschreibung:** Diese Methode liefert den Namen einer Variablen.

**Rückgabewerte:**

*BSTR*: Name der Variable

**Beispiel:**

```
VName = wfvariables.MyVariable.Name
```

HRESULT ID ([out, retval] BSTR \* *pVal*)

**Beschreibung:** Diese Methode liefert die Id der Variablen.

**Rückgabewerte:**

*BSTR*: Variablenid

**Beispiel:**

```
VId = wfvariables.MyVariable.Id
```

HRESULT Value ([out, retval] BSTR \* *pVal*)

**Beschreibung:** Diese Methode liefert den Wert der Variablen.

**Rückgabewerte:**

*BSTR*: Wert der Variable

**Beispiel:**

```
Value = wfvariables.MyVariable.Value
```

HRESULT Value ([in] BSTR *newVal*)

**Beschreibung:** Diese Methode setzt den Wert der Variablen.

**Parameter:**

*BSTR*: Wert der Variable

**Beispiel:**

```
Dim NewValue 'Wert

wfvariables.MyVariable.Value = NewValue
```



HRESULT VarType ([out, retval] VarTypeEnum \* *pVal*)

**Beschreibung:** Diese Methode liefert den Variablentyp. Hierbei wird lediglich zwischen Strings, Records und Listen unterschieden.

**Rückgabewerte:**

*VarTypeEnum*: Typ

**Hinweis:** Mögliche Werte

- § 0 = normale Stringvariable
- § 1 = die Variable ist eine Liste
- § 2 = die Variable ist ein Record

**Beispiel:**

```
Type_Name = wfvariables.MyVariable.VarType
```

HRESULT VarDataType ([out, retval] VarDataTypeEnum \* *pVal*)

**Beschreibung:** Diese Methode liefert den Variablentyp (ab enaio® 6.0 SPI).

**Rückgabewerte:**

*VarDataTypeEnum*: Typ

**Hinweis:** Mögliche Werte

- § 0 = String
- § 1 = String (Integer)
- § 2 = String (Float)
- § 3 = String (DateTime)
- § 4 = DateTime
- § 5 = Date
- § 6 = Time
- § 7 = Float
- § 8 = Integer
- § 9 = List
- § 10 = Record
- § 100 = unbekannter Typ

**Beispiel:**

```
Type_Name = wfvariables.MyVariable.VarDataType
```

HRESULT FieldName ([out, retval] BSTR \* *pVal*)

**Beschreibung:** Diese Methode liefert den Name des zugeordneten Feldes auf dem Datenblatt, sofern eines zugeordnet wurde. Es wird der Datenbankname des Feldes angegeben.

**Rückgabewerte:**

*BSTR*: Datenbankname des Feldes

**Beispiel:**

```
FieldnameDB = wfvariables.MyVariable.FieldName
```

HRESULT InputMode ([out, retval] long \* pVal)

**Beschreibung:** Diese Methode gibt Auskunft über die 'Beschreibbarkeit' der Variablen.

**Rückgabewerte:**

*long*: Zugriffstyp

**Hinweis:** Mögliche Werte

- § 1 = nur lesender Zugriff zulässig
- § 2 = nur schreibender Zugriff
- § 3 = lesender und schreibender Zugriff erlaubt

**Beispiel:**

```
access = wfvariables.MyVariable.InputMode
```

HRESULT CreateListItem ([out, retval] [IWFFVariable](#) \*\* ppVariable)

**Beschreibung:** Diese Methode erzeugt ein neues Objekt zum Einfügen in die Liste. Das neue Objekt kann mit AddListItem hinzugefügt werden.

**Rückgabewerte:**

*IWFFVariable*: neues Listenelement

**Beispiel:** Neues Listenelement wird erstellt und der Liste hinzugefügt.

```
'Listenobjekt erstellen
Set NewElement = wfvariables.MyList.CreateListItem 'MyList ist als
List                                              '[String]
definiert

NewElement.value = "Hier kann jetzt ein String hineingeschrieben
werden"

'Listenobjekt der Liste hinzufügen
wfvariables.MyList.AddListItem(NewElement)
```

HRESULT AddListItem ([in] [IWFFVariable](#) \* pListItem)

**Beschreibung:** Diese Methode fügt ein neues Objekt in die List ein.

**Parameter:**

*IWFFVariable*: neues Element

**Beispiel:** siehe [CreateListItem](#)

HRESULT RemoveListItem ([in] VARIANT varItem)

**Beschreibung:** Diese Methode löscht ein Listenelement.

**Parameter:**

*VARIANT*: Index des Listenelements

**Beispiel:**

```
'MyList ist als List[String] definiert
For i = 0 To wfvariables.MyList.count - 1
    If wfvariables.MyList.item(i).value = "zu löschender String" Then
        wfvariables.MyList.RemoveListItem(i)
    Exit For
End If
Next
```

HRESULT ListType ([out, retval] ListTypeEnum \* *pVal*)

**Beschreibung:** Diese Methode liefert den Listentyp.

**Rückgabewerte:**

*ListTypeEnum*: Listentyp

**Hinweis:** Mögliche Werte

§ ItUSERDEFINED = 0,

§ ItPROCESSLIST = 1,

§ ItUNKNOWN = 100

**Beispiel:**

```
ListType = wfvariables.MyList.ListType
```

HRESULT Selected ([out, retval] VARIANT\_BOOL \* *pVal*)

**Beschreibung:** Diese Methode gibt an, ob eine Listenelement selektiert/markiert ist.

**Rückgabewerte:**

*VARIANT\_BOOL*: TRUE = Listenelement markiert

**Beispiel:**

```
marked = wfvariables.MyList.item(index).Selected
```

HRESULT Selected ([in] VARIANT\_BOOL *newVal*)

**Beschreibung:** Diese Methode selektiert/markiert ein Listenelement.

**Parameter:**

*VARIANT\_BOOL*: TRUE = Listenelement markiert

**Beispiel:**

```
`als markiert gekennzeichnet  
wfvariables.MyList.item(index).Selected = TRUE  
  
`Element demarkiert  
wfvariables.MyList.item(index).Selected = FALSE
```

HRESULT InitialValue ([out, retval] BSTR \* pVal)

**Beschreibung:** Diese Methode liefert den Defaultwert der Variablen.

**Rückgabewerte:**

*BSTR*: Defaultwert

HRESULT TypeId([out,retval] BSTR \*pVal)

**Beschreibung:** Diese Methode liefert die Id des Typen der Listenelemente.

**Rückgabewerte:**

*BSTR*: Id

HRESULT ClearList()

**Beschreibung:** Diese Methode leert die Liste.

## IWFVariables Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle bildet eine Collection mit allen verfügbaren Workflowvariablen ab. Diese Collection ist für Client-Events unter dem Namen 'WFVariables' verfügbar.

Siehe auch: [IWFVariable](#)

### Öffentliche Methoden

HRESULT [Copy](#) ([out, retval] [IWFVariables](#) \*\*ppCopy)

### Dokumentation der Elementfunktionen

HRESULT Copy ([out, retval] [IWFVariables](#) \*\* ppCopy)

**Beschreibung:** Diese Methode legt eine Kopie der Workflowvariablen-Liste an.

**Rückgabewerte:**

*IWFVariables*: Kopie der Liste

# Statusübersicht

## Prozessstatus

- § 0 = Unbekannt
- § 1 = Initialisierung
- § 2 = Prozess läuft
- § 4 = Prozess ist angehalten
- § 8 = Prozess ist aktiv
- § 16 = Prozess wurde abgebrochen
- § 32 = Prozess ist beendet
- § 16 = Prozess wurde vom System unterbrochen

## Aktivitätsstatus

- § 0 = Unbekannt
- § 1 = Angelegt
- § 2 = Gestartet
- § 4 = Ausführung des StartActivity-Events
- § 8 = Ausführung des EndActivity-Events
- § 16 = Überprüfung der Schleifenbedingung
- § 32 = Im Schleifendurchlauf
- § 64 = Arbeitsschritt angelegt
- § 128 = Arbeitsschritt personalisiert
- § 256 = Warte (Sperrfrist)
- § 512 = Warte (Subflow)
- § 1024 = Ausgeführt
- § 2048 = Weitergeleitet
- § 4096 = Abgeschlossen
- § 8192 = Angehalten
- § 16384 = Beendet
- § 268435456 = Fehler

## Codebeispiele

### Email senden (Server-Event)

```
Dim MyObjectId 'Objektid aus dem Workflowmodell
Dim from 'Absender
Dim subject 'Betreffzeile
Dim text 'Mailinhalt
Dim file 'Anhang

'Objekt des Empfängers erstellen
Set addressee = wfOrganisation.GetOrgObjectById(MyObjectId)

'Emailadresse holen
address = addressee.GetAttributeValueByName("E-Mail")

sendmessage = wfSystem.SendMail(address, from, subject, text, file)
```

### Neues Listenelement erstellen (Server-Event)

```
'Listenobjekt erstellen
Set NewElement = MyList.CreateListItem 'MyList ist als List [String]
                                         'definiert

NewElement.value = "Hier kann jetzt ein String hineingeschrieben werden"

'Listenobjekt der Liste hinzufügen
ListId = MyList.AddListItem(NewElement)
```



## Bearbeiter einer Aktivität festlegen (Server-Event)

```
Dim MyActivityId    `diese Aktivität soll die Person bearbeiten
Dim MyAdvisorId     `Id des Bearbeiters

thisProcess.SetActivityPerformer MyActivityId, MyAdvisorId
```

## Prozessverantwortlichen löschen (Server-Event)

Löschen eines Prozessverantwortlichen (in diesem Fall der Prozessersteller)

```
Set MyResponsibles = thisprocess.Responsibles

For i = 0 To MyResponsibles.count - 1
    If MyResponsibles.item(i).id = thisprocess.Creator Then
        MyResponsibles.remove(i)
        thisprocess.Responsibles = MyResponsibles
        Exit For
    End If
Next
```

## Auswahlliste an Maskentabellenfeldern dynamisch füllen (Client-Event)

Ein Wert (MyNewValue) wird in die Auswahlliste eingefügt.

```
Dim MaskfieldName      'Name des Maskenfelds in Tabellenform
Dim MyPosition          'Position der Spalten
Dim MyPropertyName     'Name der Eigenschaft
Dim MyDefaultValue     'Falls Eigenschaft nicht vorhanden, dann wird
                        'dieser Wert zurückgegeben z.B. free entry
Dim MyNewValue          'Wert, der in die Kombobox aufgenommen werden soll

MyDefaultValue = "free entry"
'Objekt des Maskenfeldes erstellen
Set Maskfield = wfmaskfields.item(MaskfieldName)
'Spaltenobjekt erstellen
Set Column = Maskfield.WFListColumns.item(MyPosition)

count = 0
Do 'ersten freien Listeneintrag suchen
    PropertyName = "Zeile" & count 'Eigenschaftsname zusammensetzen
    'Listeneinträge holen
    Value = Column.GetStringProperty(PropertyName, MyDefaultValue)

    If Value = "free entry" Then 'freien Listeneintrag gefunden

        'neuen Eintrag setzen
        Result = Column.SetStringProperty(PropertyName, MyNewValue)
        Exit Do
    End If

    count = count + 1

Loop
```

## Auswahlliste an Maskentextfeldern dynamisch füllen (Client-Event)

Ein Wert (MyNewValue) wird in die Auswahlliste eingefügt.

```
Dim MaskfieldName 'Name des Textfeldes
Dim MySectionName 'Sektionsname
Dim MyPropertyName 'Zeilenummer
Dim MyDefaultValue 'Vorbelegung
Dim MyNewValue 'Neuer Wert

MyDefaultValue = "free entry"

Set Maskfield = wfmaskfields.item(MaskfieldName)

MySectionName = Maskfield.SectionName

Set MaskfieldListe = Maskfield.WFFieldValues

count = 0
Do
    MyPropertyName = "Zeile" & count

    Value = MaskfieldListe.GetStringValue(MySectionName, _
    MyPropertyName, MyDefaultValue)

    If Value = "free entry" Then
        MaskfieldListe.WriteStringValue MySectionName, _
        MyPropertyName, MyNewValue

        Exit Do
    End If

    count = count + 1
Loop
```

## Verwendung von Serverjobs in Events

Dieses Beispielskript wird im Event(StartAktivität) des Arbeitsschrittes gestartet. Der Workflowprozess hat den Eingabeparamter 'intDoc\_ID', welcher eine Id eines DMS-Dokumentes enthält. Mit Hilfe dieser Id wird eine DMS-Recherche durchgeführt und aus den Indexdaten der Autor ermittelt. Der Autor als Teilnehmer des Arbeitsschrittes gesetzt. Weitere Informationen zu Serverjobs sind in der Jobreferenz der Server-API zu finden.

```
'für das Absetzen von Serverjobs wird die Schnittstelle oxmljsc verwendet
Set g_oxmljsc = CreateObject("oxmljsc.oxmljsc")

'Objekt mit IP-Adresse und Port des Servers initialisieren
Result = g_oxmljsc.Init(Errors, "IPAddress=Bastubbe;TCPPort=4000")

If Result = 0 Then
    'Verbindung zum Server herstellen
    Result = g_oxmljsc.Connect(Errors)

    If Result = 0 Then
        'Daten der Workstation ermitteln
        Result = g_oxmljsc.GetParameter(Errors, "StationGUID",
                                         varStationGUID)

        Result = g_oxmljsc.GetParameter(Errors, "StationIP", varStationIP)
        Result = g_oxmljsc.GetParameter(Errors, "ComputerName",
                                         varComputerName)

        varInstName = "Test"

        'Eigenschaften der Arbeitssitzung setzen
        JobName = "krn.SessionPropertiesSet"
        ParNmIn = Array("Flags", "Properties", "address", "instname",
                        "statname", "hasserveraccount")
        ParValIn = Array(0, "address;instname;statname;hasserveraccount;",
                        varStationIP & "=" & varStationGUID,
                        varInstName, varComputerName, 0)

        Result = g_oxmljsc.DoJob(Errors, JobName, ParNmIn, ParValIn,
                                FileNmIn, ParNmOut, ParValOut, FileNmOut)

        'Login durchführen (Passwort muss verschlüsselt sein)
        JobName = "krn.SessionLogin"
        ParNmIn = Array("Flags", "UserName", "UserPwd")
        ParValIn = Array(0, "Testuser", "BP1153637613716352712075471225323")

        Result = g_oxmljsc.DoJob(Errors, JobName, ParNmIn, ParValIn,
                                FileNmIn, ParNmOut, ParValOut, FileNmOut)

        'XML-String für Suche erstellen
        xml = "<DMSQuery>"
        xml = xml & "<Archive name=""Pressearchiv"">"
        xml = xml & "<ObjectType name=""Word-Texte"">"
        xml = xml & "<Fields field_schema=""DEF"">"
        xml = xml & "<Field name=""Autor""/>"
        xml = xml & "</Fields>"
```

```

xml = xml & "<Conditions>"
xml = xml & "<ObjectID>" & intDoc_ID.Value & "</ObjectID>"
xml = xml & "</Conditions>"
xml = xml & "</ObjectType>"
xml = xml & "</Archive>"
xml = xml & "</DMSQuery>"

'XML-String base64-codieren
Set cnv = CreateObject("Oxdbcpcnv.Converter")
RetVal = cnv.StringToDBPMime(xml, base64xml)

'DMS-Recherche durchführen
JobName = "dms.GetResultList"
ParNmIn = Array("Flags", "?XML")
ParValIn = Array(4096, base64xml)

Result = g_oxmljsc.DoJob(Errors, JobName, ParNmIn, ParValIn,
                        FileNmIn, ParNmOut, ParValOut, FileNmOut)

'nach Rückgabeparamter "?XML" suchen
For i = 0 To UBound (ParNmOut)
    If ParNmOut(i) = "?XML" Then
        'base64-String dekodieren
        Retval = cnv.MimeToString( ParValOut(i), xml)

        'im XML-Suchergebnis nach Wert des Feldes Autor suchen
        Set xmldoc = CreateObject("Msxml2.DOMDocument.4.0")
        Retval = xmldoc.loadxml(xml)

        xmldoc.setProperty "SelectionLanguage", "XPath"
        Set node = xmldoc.selectSingleNode

        ("//Archive/ObjectType/Rowset/Rows/Row/Value")
        name = node.text
        Exit For
    End If
Next

'Logout durchführen
JobName = "krn.SessionLogout"
ParNmIn = Array("Flags")
ParValIn = Array(0)
Result = g_oxmljsc.DoJob(Errors, JobName, ParNmIn, ParValIn,
                        FileNmIn, ParNmOut, ParValOut, FileNmOut)

'Verbindung zum Server beenden
g_oxmljsc.Disconnect (Errors)
End If

g_oxmljsc.Finalize (Errors)

End If

```

```
'die Objektid über den Namen des Autoren ermitteln und ihn als Teilnehmer
  der
'Aktivität setzen
Set List_Org_object = wfororganisation.GetOrgObjectsByName(name)
If List_Org_object.count > 0 Then
    org_obj_id = List_Org_object.item(0).Id
    thisProcess.SetActivityPerformer
    thisProcess.CurrentActivity.Activityid,

    org_obj_id
End If
```

# Index

- Absent
  - IComOrgObject 64
- AccessRights
  - IFileObject 115
- Actions
  - IComTimer 83, 84, 87
- Active
  - IOrganisation 122
- ActivePage
  - IWFCtrlPages 189, 190
- Activities
  - IComModel 44, 45
- ActivityId
  - IComFile 38
  - IComModelActivity 48
- ActivityID
  - IActiveWorkItem 17
  - IComActivity 21
- ActivityName
  - IActiveWorkItem 17
- AddFileObject
  - IWFFile 176
- AddListItem
  - IComDataField 35
  - IWFVariable 196
- Attributes
  - IComOrgObject 62
  - IOrgObject 135
- CheckPassword
  - IActiveWorkItem 17
- ClearList
  - IComDataField 37
- Copy
  - IWFVariables 199
- Count
  - IComDataField 32
  - IWFVariable 192
- CreateEmail
  - IComTimerActions 100
- CreateListItem
  - IComDataField 36
  - IWFVariable 196
- CreateRecipient
  - IComEmail 103
- CreateTimerConfig
  - IComProcess 79
- CreationActivity
  - IComFileDoc 41
- Creator
  - IComProcess 76
- cTimestampToDateTime
  - IComSystem 112
- CtrlPages
  - IWFMaskField 186
- CurrentActivity
  - IComProcess 74
- CurrentUserAbsent
  - IWFClient 171
- DataFieldType
  - IComDataField 34
- DataType
  - IWFListColumn 178
  - IWFMaskField 183
- DefaultValue
  - IComDataField 34
- Deleteable
  - IFileObject 118, 119
- DeleteFileObject
  - IWFFile 177
- Depersonalize
  - IComTimerActions 101
- DestinationTime
  - IComTimer 87
- DestinationType
  - IComTimer 88
- EmailAddress
  - IComEmailRecipient 108
- EmailForPerformers
  - IComEmail 104
- EmailForSubstitutes
  - IComEmail 105
- Emails
  - IComTimerActions 101
- Enabled
  - IWFMaskField 185
- EnablePage
  - IWFCtrlPages 190
- EndActivityId
  - IComTimer 88, 89
- EndTime
  - IComActivity 21
- FieldName
  - IWFVariable 194
- Fields
  - IWFCtrlPage 188
- File
  - IComFile 39
- FileObjects
  - IComFile 39
  - IWFFile 175
- ForbiddenObjectTypes
  - IWFFile 176
- GetActivityPerformer
  - IWFClient 171
- GetAttributeValueByName
  - IComOrgObject 66
- GetConfigCopy
  - IComTimer 89

- GetCurrentActivityById
  - IComProcess 75
- GetDataFieldByName
  - IComActivity 24
- GetFileObjectById
  - IWFFFile 177
- GetModelActivity
  - IComActivity 24
- GetOrgClassesByName
  - IOrganisation 125
- GetOrgObjectById
  - IComOrgManagement 54
  - IOrganisation 122
- GetOrgObjectsByClassName
  - IComOrgManagement 55
  - IOrganisation 123
- GetOrgObjectsByName
  - IComOrgManagement 55
  - IOrganisation 123
- GetOrgUsersRoles
  - IComOrgManagement 55
  - IOrganisation 124
- GetPredecessorsByClassName
  - IComOrgObject 63
  - IOrgObject 136
- GetPredecessorsByName
  - IComOrgObject 64
  - IOrgObject 137
- GetRActivityById
  - IComProcess 74
- GetRoles
  - IOrganisation 124
- GetStringProperty
  - IWFListColumn 179
- GetStringValue
  - IWFFFieldValues 173
- GetSuccessorsByClassName
  - IComOrgObject 63
  - IOrgObject 136
- GetSuccessorsByName
  - IComOrgObject 64
  - IOrgObject 137
- GetTimerByConfigId
  - IComProcess 79
- GetTimerByConfigName
  - IComProcess 79
- GetTimerConfigByName
  - IComProcess 80
- GetUsers
  - IOrganisation 124
- IActiveWorkItem 15
  - ActivityID 17
  - ActivityName 17
  - CheckPassword 17
  - ProcessID 16
  - ProcessName 16
  - Save 18
  - WFFFile 16
  - WFMaskFields 15

- WFVariables 16
- WorkflowID 16
- IComActivity 20
  - ActivityID 21
  - EndTime 21
  - GetDataFieldByName 24
  - GetModelActivity 24
  - Name 24
  - PerformerId 22
  - Performers 22, 23
  - RActivityId 22
  - State 25
- IComActivityList 28
- IComCollection 29
- IComDataField 30
  - AddListItem 35
  - ClearList 37
  - Count 32
  - CreateListItem 36
  - DataFieldType 34
  - DefaultValue 34
  - Id 33
  - Item 32
  - ListItemId 36
  - RemoveListItem 36
  - Selection 37
  - Value 33
- IComEmail
  - CreateRecipient 103
  - EmailForPerformers 104
  - EmailForSubstitutes 105
  - Recipients 105
  - Subject 106
  - Text 106
- IComEmailRecipient
  - EmailAddress 108
  - Type 109
  - UserId 109
- IComFile 38
  - ActivityId 38
  - File 39
  - FileObjects 39
  - ProcessId 38
- IComFileDoc 40
  - CreationActivity 41
  - Id 41
  - IsDeletable 42
  - IsMoveable 42
  - IsWFTrayObjekt 41
  - IsWorkspaceObjekt 41
  - ObjectType 41
  - OriginalDocId 42
  - UseActiveVariant 42
- IComFileDocs 43
- IComModel 44
  - Activities 44, 45
- IComModelActivity 47
  - ActivityId 48
  - Name 49



- Performers 47
- Type 49
- IComModelActivityList 50
- IComOrgClass 51
  - Id 51
  - Name 52
- IComOrgClassList 53
- IComOrgManagement 54
  - GetOrgObjectById 54
  - GetOrgObjectsByClassName 55
  - GetOrgObjectsByName 55
  - GetOrgUsersRoles 55
- IComOrgObjAttribute 56
  - Id 56
  - Name 57
  - Value 57
- IComOrgObjAttributeList 58
- IComOrgObject 59
  - Absent 64
  - Attributes 62
  - GetAttributeValueByName 66
  - GetPredecessorsByClassName 63
  - GetPredecessorsByName 64
  - GetSuccessorsByClassName 63
  - GetSuccessorsByName 64
  - Id 60
  - Name 61
  - OrgClass 62
  - Performer 65
  - Predecessors 61
  - Substitutes 65
  - Successors 61
- IComOrgObjectList 67
- IComPerformer 68
  - Id 69
  - OrgObject 69
- IComPerformerList 71
- IComProcess 72
  - CreateTimerConfig 79
  - Creator 76
  - CurrentActivity 74
  - GetCurrentActivityById 75
  - GetRActivityById 74
  - GetTimerByConfigId 79
  - GetTimerByConfigName 79
  - GetTimerConfigByName 80
  - InsertDocIntoFile 75
  - MergeProtocol 78
  - ProcessId 74
  - ProcessName 77
  - Responsibles 78
  - SetActivityPerformer 76
  - TimerConfigs 80
  - Timers 80
  - WFFile 76
  - WorkflowModel 77
  - WorkflowName 77
- IComSystem 111
  - cTimestampToDateTime 112

- SendMail 111
- IComTimer 83, 85, 96
  - IComTimer
    - Actions 83
  - IComTimer
    - Actions 83
  - IComTimer
    - Actions 83
  - IComTimer
    - Actions 84
  - IComTimer
    - Actions 87
  - IComTimer
    - DestinationTime 87
  - IComTimer
    - DestinationTime 87
  - IComTimer
    - DestinationType 88
  - IComTimer
    - DestinationType 88
  - IComTimer
    - EndActivityId 88
  - IComTimer
    - EndActivityId 89
  - IComTimer
    - GetConfigCopy 89
  - IComTimer
    - OrganisationId 89
  - IComTimer
    - ProcessId 89
  - IComTimer
    - RunAtOnce 90
  - IComTimer
    - RunAtOnce 90
  - IComTimer
    - StartActivityId 90
  - IComTimer
    - StartActivityId 91
  - IComTimer
    - StartType 91
  - IComTimer
    - StartType 91
  - IComTimer
    - State 92
  - IComTimer
    - State 92
  - IComTimer
    - TimerConfigId 92
  - IComTimer
    - TimerConfigName 93
  - IComTimer
    - TimerConfigName 93
  - IComTimer
    - TimerId 93
  - IComTimer
    - TimerType 94
  - IComTimer
    - TimerType 94
  - IComTimer

- TimeType 95
- IComTimer
  - TimeType 95
- IComTimer
  - WorkflowId 95
- IComTimer
  - TimerType 96
- IComTimer
  - TimerType 96
- IComTimer
  - TimerType 96
- IComTimer
  - TimerType 96
- IComTimer
  - TimerType 96
- IComTimer
  - TimerType 96
- IComTimer
  - TimerType 96
- IComTimer
  - TimerType 96
- IComTimer
  - TimerType 96
- IComTimer
  - TimerType 96
- IComTimer
  - TimerType 96
- IComTimer
  - TimerType 96
- IComTimer
  - TimerType 97
- IComTimer
  - TimerType 97
- IComTimer
  - TimerType 97
- IComTimer
  - TimerType 97
- IComTimer
  - TimerType 97
- IComTimer
  - TimerType 97
- IComTimer
  - TimerType 97
- IComTimer
  - TimerType 97
- IComTimer
  - TimerType 98
- IComTimerActions 100
  - CreateEmail 100
  - Depersonalize 101
  - Emails 101
  - WorkItemForSubstitutes 102
  - WorkItemPerformers 102
- Id
  - IComDataField 33
  - IComFileDoc 41
  - IComOrgClass 51
  - IComOrgObjAttribute 56
  - IComOrgObject 60
  - IComPerformer 69
  - IOrganisation 121
  - IOrgAttribute 127
  - IOrgClass 130
  - IOrgObject 133, 140, 146, 147, 148, 149,  
150, 154, 162, 163, 164

- ID
  - FileObject 114
  - IWFMaskField 186
  - IWFVariable 193
- FileObject 114
  - AccessRights 115
  - Deleteable 118, 119
  - ID 114
  - IsNewObject 117
  - IsWorkspaceObject 116
  - Location 117
  - Moveable 118
  - ObjectType 115
- FileObjects 120
- InitialValue
  - IWFVariable 198
- InputMode
  - IWFVariable 195
- InsertDocIntoFile
  - IComProcess 75
- IOrganisation 121
  - Active 122
  - GetOrgClassesByName 125
  - GetObjectById 122
  - GetOrgObjectsByClassName 123
  - GetOrgObjectsByName 123
  - GetOrgUsersRoles 124
  - GetRoles 124
  - GetUsers 124
  - Id 121
  - Name 122
- IOrgAttribute 126
  - Id 127
  - Name 126
  - Value 127
- IOrgAttributes 128
- IOrgClass 129
  - Id 130
  - Name 130
  - OrganisationId 130
- IOrgObject 132
  - Attributes 135
  - GetPredecessorsByClassName 136
  - GetPredecessorsByName 137
  - GetSuccessorsByClassName 136
  - GetSuccessorsByName 137
  - Id 133, 140, 146, 147, 148, 149, 150, 154, 162, 163, 164
  - Name 134
  - Predecessors 134
  - Substitutes 137
  - Successors 134
- IOrgObjects 138, 140, 146, 154, 156, 157, 162, 166
- IScriptResult 167
  - ResultCode 167, 168
  - ResultString 168, 169
  - ShowResultString 169
- IsDeletable

- IComFileDoc 42
- IsMoveable
  - IComFileDoc 42
- IsNewObject
  - IFileObject 117
- IsWFTrayObjekt
  - IComFileDoc 41
- IsWorkspaceObject
  - IFileObject 116
- IsWorkspaceObjekt
  - IComFileDoc 41
- Item
  - IComDataField 32
  - IWFVariable 192
- IWFClient 170
  - CurrentUserAbsent 171
  - GetActivityPerformer 171
  - SetUserAbsence 170
  - UserName 171
- IWFCtrlPage 188
  - Field 188
  - Name 188
  - PageNumber 189
- IWFCtrlPages 189
  - ActivePage 189, 190
  - EnablePage 190
  - PageEnabled 190
- IWFFieldValues 172
  - GetStringValue 173
  - WriteStringValue 173
- IWFFile 175
  - AddFileObject 176
  - DeleteFileObject 177
  - FileObjects 175
  - ForbiddenObjectTypes 176
  - GetFileObjectByID 177
- IWFListColumn 178
  - DataType 178
  - GetStringProperty 179
  - Name 178
  - SetStringProperty 180
- IWFListColumns 181
  - TableName 181
- IWFMaskField 182
  - CtrlPages 186
  - DataType 183
  - Enabled 185
  - ID 186
  - Name 183
  - SectionName 184
  - Visible 184, 185
  - WFFieldValues 182
  - WFListColumns 184
- IWFMaskFields 187
- IWFVariable 190
  - AddListItem 196
  - Count 192
  - CreateListItem 196
  - FieldName 194

- ID 193
- InitialValue 198
- InputMode 195
- Item 192
- ListType 197
- Name 192
- RemoveListItem 196
- Selected 197
- Value 193
- VarType 194
- IWFVariables 199
  - Copy 199
- ListItemId
  - IComDataField 36
- ListType
  - IWFVariable 197
- Location
  - IFileObject 117
- MergeProtocol
  - IComProcess 78
- Moveable
  - IFileObject 118
- Name
  - IComActivity 24
  - IComModelActivity 49
  - IComOrgClass 52
  - IComOrgObjAttribute 57
  - IComOrgObject 61
  - IOrganisation 122
  - IOrgAttribute 126
  - IOrgClass 130
  - IOrgObject 134
  - IWFCtrlPage 188
  - IWFListColumn 178
  - IWFMaskField 183
  - IWFVariable 192
- ObjectType
  - IComFileDoc 41
  - IFileObject 115
- OrganisationId
  - IComTimer 89
  - IOrgClass 130
- OrgClass
  - IComOrgObject 62
- OrgObject
  - IComPerformer 69
- OriginalDocId
  - IComFileDoc 42
- PageEnabled
  - IWFCtrlPages 190
- PageNumber
  - IWFCtrlPage 189
- Performer
  - IComOrgObject 65
- PerformerId
  - IComActivity 22
- Performers
  - IComActivity 22, 23
  - IComModelActivity 47

Predecessors  
     IComOrgObject 61  
     IOrgObject 134  
 ProcessId  
     IComFile 38  
     IComProcess 74  
     IComTimer 89  
 ProcessID  
     IActiveWorkItem 16  
 ProcessName  
     IActiveWorkItem 16  
     IComProcess 77  
 RActivityId  
     IComActivity 22  
 Recipients  
     IComEmail 105  
 RemoveListItem  
     IComDataField 36  
     IWFFVariable 196  
 Responsibles  
     IComProcess 78  
 ResultCode  
     IScriptResult 167, 168  
 ResultString  
     IScriptResult 168, 169  
 RunAtOnce  
     IComTimer 90  
 Save  
     IActiveWorkItem 18  
 SectionName  
     IWFFMaskField 184  
 Selected  
     IWFFVariable 197  
 Selection  
     IComDataField 37  
 SendMail  
     IComSystem 111  
 SetActivityPerformer  
     IComProcess 76  
 SetStringProperty  
     IWFFListColumn 180  
 SetUserAbsence  
     IWFFClient 170  
 ShowResultString  
     IScriptResult 169  
 StartActivityId  
     IComTimer 90, 91  
 StartType  
     IComTimer 91  
 State  
     IComActivity 25  
     IComTimer 92  
 Subject  
     IComEmail 106  
 Substitutes  
     IComOrgObject 65  
     IOrgObject 137  
 Successors  
     IComOrgObject 61  
     IOrgObject 134  
 TableName  
     IWFFListColumns 181  
 Text  
     IComEmail 106  
 TimerConfigId  
     IComTimer 92  
 TimerConfigName  
     IComTimer 93  
 TimerConfigs  
     IComProcess 80  
 TimerId  
     IComTimer 93  
 Timers  
     IComProcess 80  
 TimerType  
     IComTimer 94, 96, 97, 98  
 TimeType  
     IComTimer 95  
 Type  
     IComEmailRecipient 109  
     IComModelActivity 49  
 UseActiveVariant  
     IComFileDoc 42  
 UserId  
     IComEmailRecipient 109  
 UserName  
     IWFFClient 171  
 Value  
     IComDataField 33  
     IComOrgObjAttribute 57  
     IOrgAttribute 127  
     IWFFVariable 193  
 VarType  
     IWFFVariable 194  
 Visible  
     IWFFMaskField 184, 185  
 WFFieldValues  
     IWFFMaskField 182  
 WFFFile  
     IActiveWorkItem 16  
     IComProcess 76  
 WFFListColumns  
     IWFFMaskField 184  
 WFFMaskFields  
     IActiveWorkItem 15  
 WFFVariables  
     IActiveWorkItem 16  
 WorkflowClient 13  
 WorkflowEngine 11  
 WorkflowId  
     IComTimer 95  
 WorkflowID  
     IActiveWorkItem 16  
 WorkflowModel  
     IComProcess 77  
 WorkflowName  
     IComProcess 77  
 WorkItemForSubstitutes

IComTimerActions 102  
WorkItemPerformers  
    IComTimerActions 102

WriteStringValue  
    IWFFieldValues 173