

# Softwaredokumentation

## COM-Schnittstellen für den enaio® Workflow-Client

Version 10.10

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 Vertreiben der Software stellt Diebstahl geistigen Eigentums dar und wird strafrechtlich verfolgt. Alle Rechte vorbehalten, einschließlich der Wiedergabe, Übermittlung, Übersetzung sowie Speicherung mit/auf Medien aller Art. Für vorkonfigurierte Testszenarien oder Demo-Präsentationen gilt: Alle Firmennamen und Personen, die in Beispielen (Screenshots) erscheinen, sind frei erfunden. Eventuelle Ähnlichkeiten mit tatsächlich existierenden Firmen und Personen sind zufällig und unbeabsichtigt.

Copyright 1992 – 2023 by

OPTIMAL SYSTEMS GmbH  
Cicerostraße 26  
D-10709 Berlin

08.05.2023  
Version 10.10

# Inhalt

Einleitung	5
Zweck .....	5
Begriffserklärung .....	5
Verbindung zum Workflowsystem .....	5
Schnittstellenübersicht des Workflow-Client .....	6
Ausführen von VB-Scriptcode mit Hilfe von axwfsript .....	7
Klassen-Verzeichnis	8
Klassen-Dokumentation	13
_IWFClientEvents Schnittstellenreferenz .....	13
Ausführliche Beschreibung .....	13
Öffentliche Methoden .....	13
Dokumentation der Elementfunktionen .....	13
IActiveWorkItem Schnittstellenreferenz.....	15
Ausführliche Beschreibung .....	15
Öffentliche Methoden .....	15
Dokumentation der Elementfunktionen .....	15
IFileObject Schnittstellenreferenz .....	20
Ausführliche Beschreibung .....	20
Öffentliche Methoden .....	20
Dokumentation der Elementfunktionen .....	20
IFileObjects Schnittstellenreferenz .....	24
Ausführliche Beschreibung .....	24
IOrganisation Schnittstellenreferenz.....	25
Ausführliche Beschreibung .....	25
Öffentliche Methoden .....	25
Dokumentation der Elementfunktionen .....	25
IOrgAttribute Schnittstellenreferenz.....	30
Ausführliche Beschreibung .....	30
Öffentliche Methoden .....	30
Dokumentation der Elementfunktionen .....	30
IOrgAttributes Schnittstellenreferenz .....	31
Ausführliche Beschreibung .....	31
IOrgClass Schnittstellenreferenz .....	32
Ausführliche Beschreibung .....	32
Öffentliche Methoden .....	32
Dokumentation der Elementfunktionen .....	33
IOrgClasses Schnittstellenreferenz .....	34
Ausführliche Beschreibung .....	34
IOrgObject Schnittstellenreferenz .....	35
Ausführliche Beschreibung .....	35
Öffentliche Methoden .....	35
Dokumentation der Elementfunktionen .....	36
IOrgObjects Schnittstellenreferenz .....	39
Ausführliche Beschreibung .....	39
IRoutingLists Schnittstellenreferenz .....	40
Ausführliche Beschreibung .....	40
IRoutingList Schnittstellenreferenz.....	41
Ausführliche Beschreibung .....	41
IRoutingListEntry Schnittstellenreferenz.....	47
IRoutingListItem Schnittstellenreferenz .....	48

IRoutingListItemParticipants Schnittstellenreferenz .....	54
IAdhocReminderPeriodTemplate Schnittstellenreferenz .....	55
IAdhocReminderPeriodTemplates Schnittstellenreferenz.....	57
IScriptResult Schnittstellenreferenz .....	58
Ausführliche Beschreibung .....	58
Öffentliche Methoden .....	58
Dokumentation der Elementfunktionen .....	58
IWFActivities Schnittstellenreferenz .....	61
Ausführliche Beschreibung .....	61
IWFActivity Schnittstellenreferenz.....	62
Ausführliche Beschreibung .....	62
Öffentliche Methoden .....	62
Dokumentation der Elementfunktionen .....	62
IWFActivities Schnittstellenreferenz .....	67
Ausführliche Beschreibung .....	67
IWFActivity Schnittstellenreferenz .....	68
Ausführliche Beschreibung .....	68
Öffentliche Methoden .....	68
Dokumentation der Elementfunktionen .....	68
IWFActivityPerformer Schnittstellenreferenz .....	72
Ausführliche Beschreibung .....	72
Öffentliche Methoden .....	72
Dokumentation der Elementfunktionen .....	72
IWFActivityPerformers Schnittstellenreferenz.....	73
Ausführliche Beschreibung .....	73
IWFHistActivities Schnittstellenreferenz .....	74
Ausführliche Beschreibung .....	74
IWFHistActivity Schnittstellenreferenz.....	75
Ausführliche Beschreibung .....	75
Öffentliche Methoden .....	75
Dokumentation der Elementfunktionen .....	75
IWFHistEntries Schnittstellenreferenz .....	76
Ausführliche Beschreibung .....	76
IWFHistEntry Schnittstellenreferenz .....	77
Ausführliche Beschreibung .....	77
Öffentliche Methoden .....	77
Dokumentation der Elementfunktionen .....	77
IWFHistTimer Schnittstellenreferenz .....	81
Ausführliche Beschreibung .....	81
Öffentliche Methoden .....	81
Dokumentation der Elementfunktionen .....	81
IWFHistTimers Schnittstellenreferenz.....	84
Ausführliche Beschreibung .....	84
IWFHistWorkflow Schnittstellenreferenz.....	85
Ausführliche Beschreibung .....	85
Öffentliche Methoden .....	85
Dokumentation der Elementfunktionen .....	85
IWFHistWorkflows Schnittstellenreferenz .....	87
Ausführliche Beschreibung .....	87
IWFHistWorkItemRelEntries Schnittstellenreferenz .....	88
Ausführliche Beschreibung .....	88
IWFHistWorkItemRelEntry Schnittstellenreferenz.....	89
Ausführliche Beschreibung .....	89
Öffentliche Methoden .....	89
Dokumentation der Elementfunktionen .....	89
IWFAModel Schnittstellenreferenz.....	93
Ausführliche Beschreibung .....	93
Öffentliche Methoden .....	93
Dokumentation der Elementfunktionen .....	93

IWFAModels Schnittstellenreferenz .....	94
Ausführliche Beschreibung .....	94
IWFAProcess Schnittstellenreferenz .....	95
Ausführliche Beschreibung .....	95
Öffentliche Methoden .....	95
Dokumentation der Elementfunktionen .....	96
IWFAProcesses Schnittstellenreferenz .....	105
Ausführliche Beschreibung .....	105
IWFAProcessLock Schnittstellenreferenz .....	106
Ausführliche Beschreibung .....	106
Öffentliche Methoden .....	106
Dokumentation der Elementfunktionen .....	106
IWFAProcessLocks Schnittstellenreferenz .....	108
Ausführliche Beschreibung .....	108
IWFALockInfoEntry Schnittstellenreferenz .....	109
Ausführliche Beschreibung .....	109
Öffentliche Methoden .....	109
Dokumentation der Elementfunktionen .....	109
IWFALockInfoEntries Schnittstellenreferenz .....	111
Ausführliche Beschreibung .....	111
IWFARole Schnittstellenreferenz .....	112
Ausführliche Beschreibung .....	112
Öffentliche Methoden .....	112
Dokumentation der Elementfunktionen .....	112
IWFARoles Schnittstellenreferenz .....	113
Ausführliche Beschreibung .....	113
IWFASUser Schnittstellenreferenz .....	114
Ausführliche Beschreibung .....	114
Öffentliche Methoden .....	114
Dokumentation der Elementfunktionen .....	114
IWFASUsers Schnittstellenreferenz .....	115
Ausführliche Beschreibung .....	115
IWFASWorkerqueueEntries Schnittstellenreferenz .....	116
Ausführliche Beschreibung .....	116
IWFASWorkerqueueEntry Schnittstellenreferenz .....	117
Ausführliche Beschreibung .....	117
Öffentliche Methoden .....	117
Dokumentation der Elementfunktionen .....	117
IWFClient Schnittstellenreferenz .....	120
Ausführliche Beschreibung .....	120
Öffentliche Methoden .....	120
Dokumentation der Elementfunktionen .....	122
IWFEvent Schnittstellenreferenz .....	138
Ausführliche Beschreibung .....	138
Öffentliche Methoden .....	138
Dokumentation der Elementfunktionen .....	138
IWFEvents Schnittstellenreferenz .....	140
Ausführliche Beschreibung .....	140
Öffentliche Methoden .....	140
Dokumentation der Elementfunktionen .....	140
IWFFieldValues Schnittstellenreferenz .....	141
Ausführliche Beschreibung .....	141
Öffentliche Methoden .....	141
Dokumentation der Elementfunktionen .....	142
IWFFile Schnittstellenreferenz .....	144
Ausführliche Beschreibung .....	144
Öffentliche Methoden .....	144
Dokumentation der Elementfunktionen .....	144
IWFItem Schnittstellenreferenz .....	146

Ausführliche Beschreibung .....	146
Öffentliche Methoden .....	146
Dokumentation der Elementfunktionen .....	146
IWFItems Schnittstellenreferenz .....	149
Ausführliche Beschreibung .....	149
IWFListColumn Schnittstellenreferenz .....	150
Ausführliche Beschreibung .....	150
Öffentliche Methoden .....	150
Dokumentation der Elementfunktionen .....	150
IWFListColumns Schnittstellenreferenz .....	154
Ausführliche Beschreibung .....	154
Öffentliche Methoden .....	154
Dokumentation der Elementfunktionen .....	154
IWFMaskField Schnittstellenreferenz .....	155
Ausführliche Beschreibung .....	155
Öffentliche Methoden .....	155
Dokumentation der Elementfunktionen .....	155
IWFMaskFields Schnittstellenreferenz.....	158
Ausführliche Beschreibung .....	158
IWFCtrlPage Schnittstellenreferenz .....	159
Ausführliche Beschreibung .....	159
IWFCtrlPages Schnittstellenreferenz.....	160
Ausführliche Beschreibung .....	160
IWFOrganisation Schnittstellenreferenz.....	162
Ausführliche Beschreibung .....	162
Öffentliche Methoden .....	162
Dokumentation der Elementfunktionen .....	162
IWFOrganisations Schnittstellenreferenz .....	163
Ausführliche Beschreibung .....	163
IWFScript Schnittstellenreferenz.....	164
Ausführliche Beschreibung .....	164
Öffentliche Methoden .....	164
Dokumentation der Elementfunktionen .....	164
IWFVariable Schnittstellenreferenz .....	165
Ausführliche Beschreibung .....	165
Öffentliche Methoden .....	165
Dokumentation der Elementfunktionen .....	166
IWFVariables Schnittstellenreferenz .....	171
Ausführliche Beschreibung .....	171
Öffentliche Methoden .....	171
Dokumentation der Elementfunktionen .....	171
IWFWorkItem Schnittstellenreferenz.....	172
Ausführliche Beschreibung .....	172
Öffentliche Methoden .....	172
Dokumentation der Elementfunktionen .....	172
IWFWorkItems Schnittstellenreferenz .....	178
Ausführliche Beschreibung .....	178
Statusübersicht .....	179
Prozessstatus.....	179
Aktivitätsstatus .....	179
Codebeispiele .....	180
Anmeldung am enaio® und Verbindung zum Workflow (VBS) .....	180
Verwendung von Serverjobs und Start eines Workflows aus axadmin.exe .....	181
Programmierbeispiel in Visual Basic.....	184



# Einleitung

## Zweck

Dieses Dokument enthält die Dokumentation der COM-Schnittstellen für den enaio® – Workflow Client. Mithilfe dieser Komponenten ist der Zugriff auf die entsprechenden Funktionalitäten aus z.B. einer VB, VBS oder auch C++ Umgebung möglich.

## Begriffserklärung

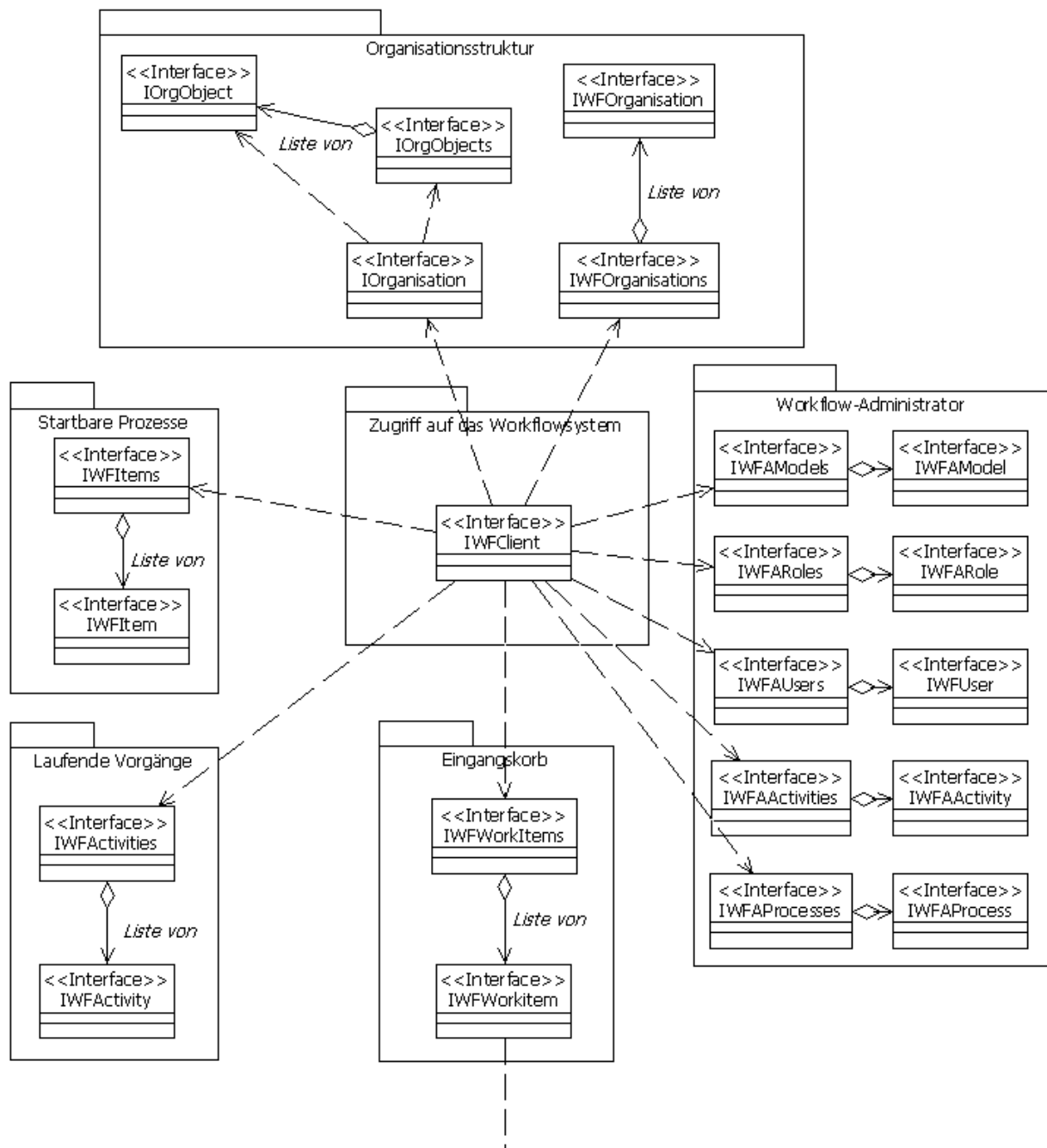
Workflowmodelle automatisieren Geschäftsvorfälle. Sie bestehen aus einer logischen Anordnung von Tätigkeiten (Aktivitäten), den zu bearbeitenden Daten (Workflowvariablen, Dokumente) und den Personen, welche die Tätigkeiten ausführen. Die Modellierung eines Workflowmodells erfolgt graphisch über den enaio® - Workflow Editor. Mit Hilfe eines Workflowmodells wird durch eine Clientanwendung ein Prozess erstellt, welcher eine konkrete Instanz des Modells bzw. Geschäftsvorfalles 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 Workflowengine 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 Skriptcode (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 dem enaio® - Workflow Client 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.

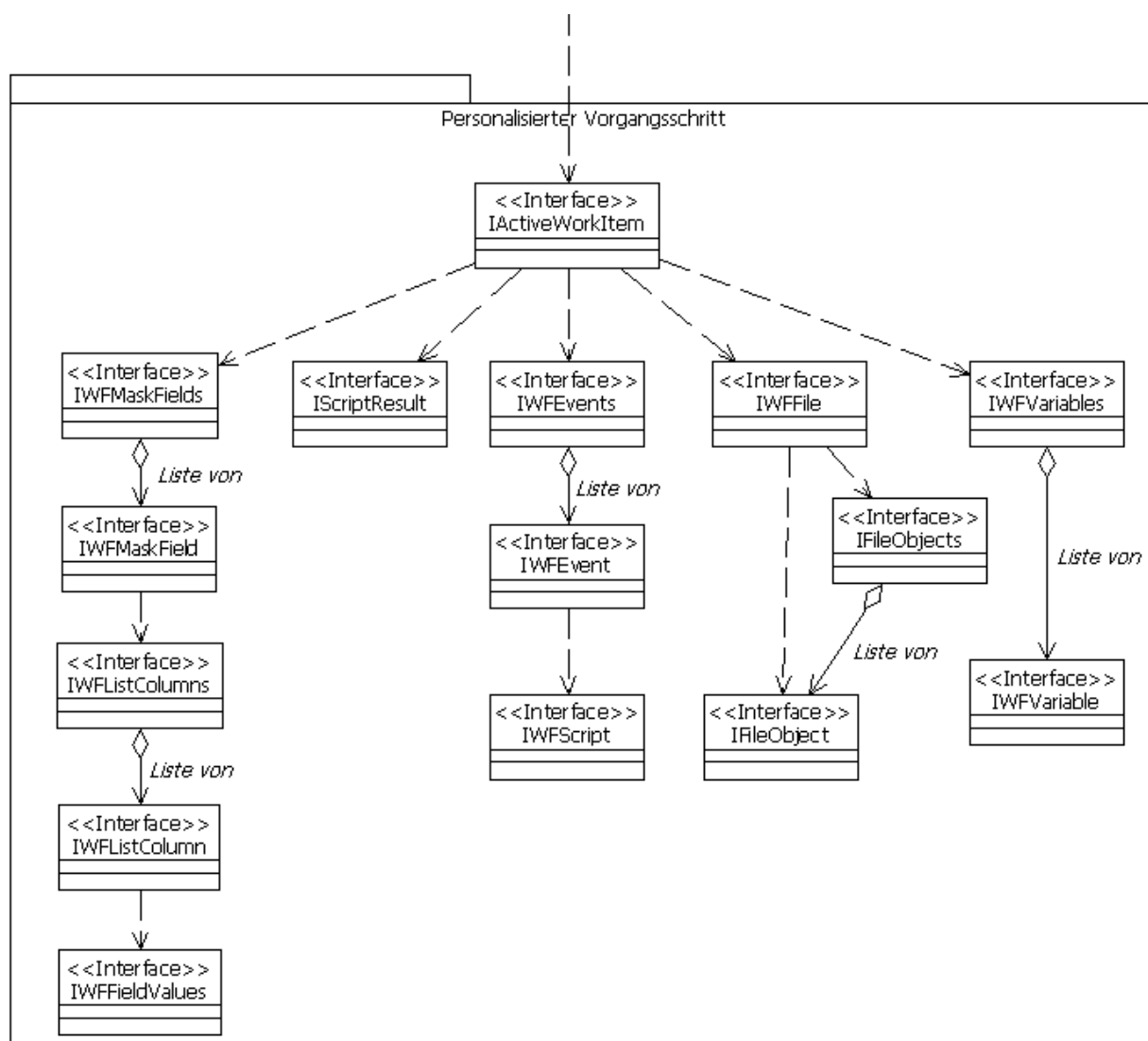
## Verbindung zum Workflowsystem

Damit alle Schnittstellen korrekt arbeiten, muss als erster Arbeitsschritt eine Verbindung zum Workflowsystem hergestellt werden. Eine Verbindung kann nur für Benutzer, die im Workflowsystem bekannt sind, hergestellt werden. Workflow-Benutzer können im Workfloweditor definiert werden. Die Schnittstelle [IWFCClient](#) stellt zum Verbindungsaufbau die Methode Connect bereit. Diese Methode setzt eine Anmeldung am enaio® voraus. Die Anmeldung am enaio® kann über die Schnittstelle ASSystem (AXBASICS.DLL) erfolgen. Wenn keine Anmeldung über diese Schnittstelle erfolgt ist, übernimmt das die Methode [IWFCClient::Connect\(\)](#). Damit ein Connect erfolgreich durchgeführt werden kann, muss der anzumeldende Benutzer in der Organisationsstruktur eingetragen sein. Dies geschieht über den enaio® – Workflow Editor. Nach einem erfolgreichen Connect stehen dem Benutzer nun die Funktionalitäten des Workflowsystems zur Verfügung. So können nun Eingangskörbe abgefragt werden, Prozesse gestartet und Arbeitsschritte bearbeitet werden. Wenn die jeweilige Umgebung dies erlaubt, können über die Schnittstelle [IWFCClientEvents](#) besondere Events des Workflowsystems empfangen werden (z.B. wenn sich Änderungen im Eingangskorb ergeben haben).



## Schnittstellenübersicht des Workflow-Client





## Ausführen von VB-Skriptcode mit Hilfe von axwfsript

Zum Ausführen von VB-Skriptcode steht das OS-Tool axwfsript zur Verfügung. Das Programm hat folgende Aufrufparameter "axwfsript -s <Skriptpfad> -u <Benutzer>@<Passwort> [-q <keine Ausgabe>] [-t <Intervall in s>]". Die Anmeldung ans enaio® - System und die Verbindung zum Workflowsystem erledigt das Programm axwfsript. Das Programm stellt dem Skriptcode ein Objekt wfclient bereit, über welches die Schnittstelle [IWFCClient](#) angesprochen werden kann.

## Klassen-Verzeichnis

Hier folgt die Aufzählung aller Schnittstellen mit einer Kurzbeschreibung:

Bereich	Klassen	Beschreibung
Zugriff auf das Workflowsystem	<a href="#"><u>IWFClient</u></a>	Schnittstelle ermöglicht Zugriff auf das Workflowsystem (in den Client-Events unter dem Namen 'wfclient' verfügbar).
Organisationsstruktur	<a href="#"><u>IOrganisation</u></a>	Schnittstelle beschreibt die Organisationsstruktur (in den Client-Events unter dem Namen 'wforganisation' verfügbar).
	<a href="#"><u>IOrgClasses</u></a>	Liste von Organisationsklassen
	<a href="#"><u>IOrgClass</u></a>	Schnittstelle beschreibt eine Organisationsklasse
	<a href="#"><u>IOrgObjects</u></a>	Liste von Organisationsobjekten.
	<a href="#"><u>IOrgObject</u></a>	Schnittstelle beschreibt ein Organisationsobjekt.
	<a href="#"><u>IOrgAttributes</u></a>	Liste von Objektattributen.
	<a href="#"><u>IOrgAttribute</u></a>	Schnittstelle beschreibt ein Objektattribut.
	<a href="#"><u>IWFOrganisation</u></a>	Schnittstelle beschreibt eine Organisation.
	<a href="#"><u>IWFOrganisations</u></a>	Liste aller Organisationen.
Startbare Vorgänge	<a href="#"><u>IWFItems</u></a>	Liste aller startbaren Vorgänge.
	<a href="#"><u>IWFItem</u></a>	beschreibt einen startbaren Vorgang
Startbare Vorgangsschritte	<a href="#"><u>IWFWorkItems</u></a>	Liste aller Vorgangsschritte im Eingangskorb.
	<a href="#"><u>IWFWorkItem</u></a>	Schnittstelle beschreibt ein startbaren Vorgangsschritt.

Personalisierter Vorgangsschritt	Vorgangsschritt	<a href="#"><u>IActiveWorkItem</u></a>	Schnittstelle beschreibt personalisierten Vorgangsschritt (in den Client-Events unter dem Namen 'Workitem' verfügbar).
	Akte und Dokument	<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.
		<a href="#"><u>IFileObject</u></a>	Schnittstelle liefert Informationen zum Dokument aus dem enaio® - System.
	Event und Skript	<a href="#"><u>IWFEvents</u></a>	Liste von Events.
		<a href="#"><u>IWFEvent</u></a>	Schnittstelle beschreibt einen Workflow-Event.
		<a href="#"><u>IWFScript</u></a>	Schnittstelle für ein Skript zum Event.
		<a href="#"><u>IScriptResult</u></a>	Programmverhalten kann bestimmt werden (in den Client-Events unter dem Namen 'ScriptResult' verfügbar).
	Maskenfelder und Variablen	<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 Maskenfelds.
		<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.
		<a href="#"><u>IWFFieldValues</u></a>	Schnittstelle beschreibt Auswahlliste an Textfeldern in Masken.
		<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.
	Adhoc	<a href="#"><u>IRoutingLists</u></a>	Liste mit Elementen vom Typ IRoutingList.
		<a href="#"><u>IRoutingList</u></a>	Die Laufliste, wenn der Arbeitsschritt ein Adhoc-Arbeitsschritt ist. Kann auch eine Vorlage für eine Laufliste repräsentieren.
		<a href="#"><u>IRoutingListEntry</u></a>	Ein Eintrag in der Laufliste eines Adhoc-Arbeitsschrittes.
		<a href="#"><u>IRoutingListItem</u></a>	Ein Element eines Eintrags in der Laufliste eines Adhoc-Arbeitsschrittes.

		<a href="#"><u>IAdhocReminderPeriodTemplates</u></a>	Eine Liste mit Vorlagen für Mahnfristen, die für RoutingListItems verwendet werden kann.
		<a href="#"><u>IAdhocReminderPeriodTemplate</u></a>	Eine Vorlage für eine Mahnfrist, die für RoutingListItems verwendet werden kann.

Laufende Vorgänge		<a href="#">IWFAActivities</a>	Liste aller laufenden Aktivitäten.
		<a href="#">IWFAActivity</a>	Schnittstelle beschreibt eine laufende Aktivität.
Sonstiges		<a href="#">IWFCClientEvents</a>	Eventschnittstelle des Workflowsystems
		<a href="#">IWFAActivityPerformers</a>	Liste aller Teilnehmer einer Aktivität
		<a href="#">IWFAActivityPerformer</a>	Schnittstelle beschreibt einen Teilnehmer-/rolle einer Aktivität
Schnittstellen für den enaio® - Workflow Administrator	Adminis- tration	<a href="#">IWFAModels</a>	Liste von Workflowmodellen
		<a href="#">IWFAModel</a>	Schnittstelle beschreibt ein Workflowmodell
		<a href="#">IWFAProcesses</a>	Liste von Prozessen
		<a href="#">IWFAProcess</a>	Schnittstelle beschreibt einen Prozess
		<a href="#">IWFAActivities</a>	Liste aller Vorgangsschritte eines Prozesses
		<a href="#">IWFAActivity</a>	Schnittstelle beschreibt einen Vorgangsschritt
		<a href="#">IWFARoles</a>	Liste von Rollen
		<a href="#">IWFARole</a>	Schnittstelle beschreibt eine Rolle
		<a href="#">IWFAUsers</a>	Liste von Benutzern
		<a href="#">IWFAUser</a>	beschreibt einen Benutzer
		<a href="#">IWFAWorkerqueueEntries</a>	Liste mit Informationen zu Workerqueue-Elementen
		<a href="#">IWFAWorkerqueueEntry</a>	beschreibt Workerqueue-Element
		<a href="#">IWFAProcessLocks</a>	Liste mit Informationen zu gelockten Workflow-Prozessen
		<a href="#">IWFAProcessLock</a>	beschreibt einen vom System gelockten Prozess
		<a href="#">IWFALockInfoEntries</a>	Liste mit Informationen zu Workflow-Datenbanktabellen die vom System gelockt werden
		<a href="#">IWFALockInfoEntry</a>	beschreibt eine Workflow-Datenbanktabelle

	Historien- verwaltung	<a href="#"><u>IWFaHistWorkflows</u></a>	Liste von hist. Workflowmodellen
		<a href="#"><u>IWFaHistWorkflow</u></a>	Schnittstelle beschreibt ein hist. Workflowmodell
		<a href="#"><u>IWFaHistActivities</u></a>	Liste von hist. Vorgangsschritten.
		<a href="#"><u>IWFaHistActivity</u></a>	Schnittstelle beschreibt einen hist. Vorgangsschritt
		<a href="#"><u>IWFaHistWorkItemRelEntries</u></a>	Liste von hist. Eingangskorbeinträgen
		<a href="#"><u>IWFaHistWorkItemRelEntry</u></a>	Schnittstelle beschreibt einen hist. Eintrag für den Eintragskorb
		<a href="#"><u>IWFaHistEntries</u></a>	Liste von hist. Einträgen
		<a href="#"><u>IWFaHistEntry</u></a>	Schnittstelle beschreibt einen hist. Eintrag
		<a href="#"><u>IWFaHistTimers</u></a>	Liste von hist. Mahn-/Sperrfristen
		<a href="#"><u>IWFaHistTimer</u></a>	Schnittstelle beschreibt eine hist. Mahn-/Sperrfrist

# Klassen-Dokumentation

## \_IWFCClientEvents Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt die Events des Workflowsystems. Über diese werden Benachrichtigungen des Servers an die Clientumgebung weitergegeben. Dies geschieht z.B. immer dann, wenn sich der Eingangskorb eines Benutzers ändert.

**Hinweis:** Diese Funktionalität steht bisher nur unter C++ zur Verfügung.

### Öffentliche Methoden

**HRESULT [OnWorkItemListChanged](#) ()**  
**HRESULT [OnWorkItemWarning](#) ([in]BSTR bstrID)**  
**HRESULT [OrganisationChanged](#) ([in] BSTR bstrID)**  
HRESULT StartWizardApplication([in] IWFWorkItem\* pWorkItem)  
HRESULT AddInboxItems([in] IWFWorkItems\* pWorkItems)  
HRESULT RemoveInboxItem([in] BSTR bstrWorkItem)

### Dokumentation der Elementfunktionen

#### **HRESULT [OnWorkItemListChanged](#) ()**

**Beschreibung:** Dieses Event wird ausgelöst, wenn Änderungen im Eingangskorb vorliegen. Als Reaktion darauf sollte man den Eingangskorb neu abrufen.

Siehe: [IWFCClient::WFWorkItems\(\)](#)

#### **HRESULT [OnWorkItemWarning](#) ([in] BSTR *bstrID*)**

**Beschreibung:** Dieses Event wird ausgelöst, wenn die Mahnfrist eines Vorgangs überschritten wurde.

**Parameter:**

*BSTR*: ID des betreffenden Vorgangs

#### **HRESULT [OrganisationChanged](#) ([in] BSTR *bstrID*)**

**Beschreibung:** Dieses Event wird ausgelöst, wenn im enaio® – Workflow Editor eine andere Organisation aktiviert wurde.

**Parameter:**

*BSTR*: ID der betreffenden Organisation

#### **HRESULT [StartWizardApplication](#)([in] IWFWorkItem\* pWorkItem)**

**Beschreibung:** Dieses Event wird ausgelöst, wenn für den Benutzer ein Arbeitsschritt vorliegt, der automatisch geöffnet werden soll.

**Parameter:**



IWFWorkItem: Arbeitsschritt

### **HRESULT AddInboxItems([in] IWFWorkItems\* pWorkItems)**

**Beschreibung:** Dieses Event wird ausgelöst, wenn für den Benutzer neue Arbeitsschritte vorliegen. Die Arbeitsschritte werden direkt mit dem Event geliefert. Das Event wird auch ausgelöst, wenn sich an einem Arbeitsschritt etwas geändert hat.

**Parameter:**

IWFWorkItems: Neue Arbeitsschritte.

### **HRESULT RemoveInboxItem([in] BSTR bstrWorkItem)**

**Beschreibung:** Dieses Event wird ausgelöst, wenn aus dem Eingangskorb des Benutzers ein Arbeitsschritt entfernt wird.

**Parameter:**

*BSTR*: ID des Arbeitsschrittes

## IActiveWorkItem Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt einen personalisierten Vorgangsschritt (Aktivität). Dieser ist einem Benutzer zugeordnet und kann nur noch von diesem bearbeitet werden. In Client-Events ist das Objekt unter dem Namen 'Workitem' verfügbar. Zugriff auf diese Schnittstelle über:

- [IWFWWorkItem::Execute](#) ()

### Öffentliche Methoden

**HRESULT [Abort](#)** ([out] VARIANT \*pvarErrorString,[out, retval] long \*lpResult)  
**HRESULT [ActivityID](#)** ([out, retval] BSTR \*pVal)  
**HRESULT [ActivityName](#)** ([out, retval] BSTR \*pVal)  
**HRESULT [CheckPassword](#)** ([in] VARIANT\_BOOL newVal)  
**HRESULT [CheckPassword](#)** ([out, retval] VARIANT\_BOOL \*pVal)  
**HRESULT [Complete](#)** ([out] VARIANT \*pvarErrorString,[out, retval]long \*lpResult)  
**HRESULT [ExecuteEvent](#)** ([in] WFEEventTypeEnum etEventType,[in] long IHWND, BSTR bstrParam,[out, retval] [IScriptResult](#) \*\*ppScriptResult)  
**HRESULT [ProcessID](#)** ([out, retval] BSTR \*pVal)  
**HRESULT [ProcessName](#)** ([out, retval] BSTR \*pVal)  
**HRESULT [RoutingList](#)**([out, retval] IRoutingList\* \*pVal)  
**HRESULT [Save](#)** ([out] VARIANT \*pvarErrorString,[out, retval]long \*lpResult)  
**HRESULT [WFEEvents](#)** ([out, retval] [IWFEEvents](#) \*\*pVal)  
**HRESULT [WFFile](#)** ([out, retval] [IWFFile](#) \*\*pVal)  
**HRESULT [WFMaskFields](#)** ([out, retval] [IWFMaskFields](#) \*\*pVal)  
**HRESULT [WFVariables](#)** ([out, retval] [IWFFVariables](#) \*\*pVal)  
**HRESULT [WorkflowID](#)** ([out, retval] BSTR \*pVal)  
**HRESULT [WorkflowVersion](#)**([out, retval] long \*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 dieser Aktivität verwendeten Eingabemaske.

**Rückgabewerte:**

*IWFMaskFields:* Liste aller Maskenfelder

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

**Beschreibung:** Diese Methode liefert eine Liste aller Workflowvariablen, die dieser Aktivität zugeordnet sind. Die Werte der Workflowvariablen können hier direkt geändert werden und werden dann z.B. per [Complete\(\)](#) an die enaio® - Workflow Engine übergeben.

**Rückgabewerte:**

*IWFVariables:* Liste der Workflowvariablen

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

**Beschreibung:** Diese Methode beendet diesen Vorgangsschritt und leitet ihn weiter. Hierbei werden Änderungen an den Workflowvariablen und der Akte übernommen.

**Parameter:**

*VARIANT:* hier wird im Falle eines Fehlers eine Fehlermeldung zurückgegeben

**Rückgabewerte:**

*long:* 0 = kein Fehler

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

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

**Rückgabewerte:**

*IWFFFile:* Workflowakte

### **HRESULT WFEvents ([out, retval] [IWFEEvents](#) \*\* pVal)**

**Beschreibung:** Diese Methode liefert eine Liste aller Events die im OS:4.x-Workflow\_Editor zur Aktivität definiert wurden.

**Rückgabewerte:**

*IWFEEvents:* Liste der Events

### **HRESULT ExecuteEvent ([in] WFEEventTypeEnum etEventType, [in] long lHWND, BSTR bstrParam, [out, retval] [IScriptResult](#) \*\* ppScriptResult)**

**Beschreibung:** Diese Methode führt ein bestimmtes Event aus.

**Parameter:**

*WFEEventTypeEnum:* Typ des Events

*long:* Fenster-Handle wenn vorhanden, ansonsten 0

*BSTR:* zusätzlicher Parameter, bei Event ButtonClick sollte der Name des Buttons übergeben werden

#### Rückgabewerte:

*IScriptResult*: Objekt zur Auswertung des Scriptergebnisses

#### Hinweis: Eventtypen

- Unknown = 0 (Unbekannt)
- BeforeForward = 1 (Vor dem Weiterleiten)
- AfterForward = 2 (Nach dem Weiterleiten)
- BeforeForwardTo = 3 (Vor 'Weiterleiten an...')
- ButtonClick = 4 (Button gedrückt)
- BeforeOpen = 5 (Vor dem Öffnen der Datenmaske)
- AfterSignature = 6 (nach digitaler Signatur eines Vorgangsschrittes)
- BeforeCancel = 7 (Vor dem Abbrechen)
- SimulateMaskEdit = 8

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

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

#### Rückgabewerte:

*BSTR*: Modell-Id

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

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

#### Rückgabewerte:

*LONG*: Modell-Version

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

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

#### Rückgabewerte:

*BSTR*: Prozess-ID

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

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

#### Rückgabewerte:

*BSTR*: Prozessname

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

**Beschreibung:** Diese Methode liefert die Adhocarbeitsliste bei Adhocarbeitschritten, bei Nicht-Adhocarbeitschritten wird null zurückgegeben.

#### Rückgabewerte:

[\*IRoutingList\*](#): Prozessname

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

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

**Rückgabewerte:**

*BSTR*: Name der Aktivität

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

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

**Rückgabewerte:**

*BSTR*: ID der Aktivität

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

**Beschreibung:** Diese Methode bricht die Bearbeitung eines Vorgangsschrittes ab und speichert den aktuellen Zustand (Workflowvariablen und der Akte). Der Vorgangsschritt bleibt weiterhin personalisiert. Die Methode kann nicht während der Ausführung eines Events verwendet werden.

**Parameter:**

*VARIANT*: hier wird ggf. der Fehlertext zurückgeliefert

**Rückgabewerte:**

*long*: 0 = kein Fehler aufgetreten

### **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

### **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

### **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.

**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 zu 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 Lauflistenvorlage anhand der Id.

**Parameter:**

*BSTR*: bstrTemplateId – Id der Vorlage

**Rückgabewerte:**

[IRoutingList](#)\*: die Vorlage

## IFileObject Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt ein Dokument aus dem enaio®, Zugriff auf diese Schnittstelle über:

- [IWFFile::FileObjects\(\)](#)
- [IWFFile::GetFileObjectByID\(\)](#)

### Öffentliche Methoden

**HRESULT [AccessRights](#)** ([out, retval] DocAccessRightsEnum \*pVal)  
**HRESULT [Deleteable](#)** ([in] VARIANT\_BOOL newVal)  
**HRESULT [Deleteable](#)** ([out, retval] VARIANT\_BOOL \*pVal)  
**HRESULT [ID](#)** ([out, retval] long \*pVal)  
**HRESULT [IsNewObject](#)** ([out, retval] VARIANT\_BOOL \*pVal)  
**HRESULT [IsWorkspaceObject](#)** ([in] VARIANT\_BOOL newVal)  
**HRESULT [IsWorkspaceObject](#)** ([out, retval] VARIANT\_BOOL \*pVal)  
**HRESULT [Location](#)** ([out, retval] DocLocationEnum \*pVal)  
**HRESULT [Moveable](#)** ([in] VARIANT\_BOOL newVal)  
**HRESULT [Moveable](#)** ([out, retval] VARIANT\_BOOL \*pVal)  
**HRESULT [ObjectType](#)** ([out, retval] long \*pVal)  
**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*: ObjektID

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

**Beschreibung:** Diese Methode liefert den Typ des Dokuments in der Akte.

**Rückgabewerte:**

*long*: Objekttyp

### **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

### **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

### **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

### **HRESULT Location ([out, retval] DocLocationEnum \* pVal)**

**Beschreibung:** Diese Methode bestimmt die Position eines Dokuments im enaio®.

**Rückgabewerte:**

*DocLocationEnum*: Position

**Hinweis:** Mögliche Werte

- 1 = das Objekt liegt im enaio®
- 2 = das Objekt liegt lediglich in der Workflowablage



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

**Beschreibung:** Diese Methode gibt an, ob das Dokument neu angelegt wurde.

**Rückgabewerte:**

*VARIANT\_BOOL*: TRUE = neu angelegt

### **HRESULT Moveable ([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 Moveable ([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 Deleteable ([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 Deleteable ([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

### **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 Dokumente. Zugriff auf diese Schnittstelle über:

- [IWFFile::FileObjects\(\)](#)

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

## IOrganisation Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt die Organisationsstruktur und ist in Client-Events unter den Namen 'wforganisation' 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 Active ([out, retval] VARIANT_BOOL *pVal)
HRESULT GetOrgObjectById ([in] BSTR bstrId,[out, retval] IOrgObject
**pIOrgObject)
HRESULT GetOrgObjectsByClassName ([in] BSTR bstrClassName,[out, retval]
IOrgObjects **pIOrgObjects)
HRESULT GetOrgObjectsByName ([in] BSTR bstrName,[out, retval] IOrgObjects
**pIOrgObjects)
HRESULT GetOrgUsersRoles ([out, retval] IOrgObjects **pList)
HRESULT GetRoles ([out, retval] IOrgObjects **pList)
HRESULT GetUsers ([out, retval] IOrgObjects **pList)
HRESULT Id ([out, retval] BSTR *pVal)
HRESULT Name ([out, retval] BSTR *pVal)
HRESULT GetOrgClassById ([in] BSTR bstrId, [out,retval] IOrgClass
**pIOrgClass)
HRESULT GetOrgClassesByName ([in] BSTR bstrName, [out,retval] IOrgClasses
**pIOrgClasses)
HRESULT GetOrgObjectsByClassId ([in] BSTR bstrClassId, [out,retval]
IOrgObjects** pIOrgObjects)
HRESULT GetOrganisation ([in] BSTR ProcessID, [out,retval] IOrganisation**
pIOrganisation)
HRESULT GetProcessRoles ([out,retval] IOrgObjects** pIOrgObjects)
```

### Dokumentation der Elementfunktionen

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

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

**Rückgabewerte:**

*BSTR*: Organisationsid

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

**Beschreibung:** Diese Methode liefert den Organisationsnamen.

**Rückgabewerte:**

*BSTR*: Organisationsname

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

**Beschreibung:** Diese Methode gibt an, ob die Organisation aktiv ist.

**Rückgabewerte:**

*VARIANT\_BOOL*: TRUE = Organisation aktiv

**HRESULT GetOrgObjectById ([in] BSTR *bstrId*, [out, retval] [IOrgObject](#) \*\*  
*pIOrgObject*)**

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

**Parameter:**

*BSTR*: Objektid

**Rückgabewerte:**

*IOrgObject*: Objekt

**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

### **HRESULT GetOrgObjectsByClassName ([in] BSTR *bstrClassName*, [out, retval] [IOrgObjects](#) \*\* *pIOObjects*)**

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

**Parameter:**

*BSTR*: Klassenname der gesuchten Objekte

**Rückgabewerte:**

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

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

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

**Rückgabewerte:**

*IOrgObjects*: Liste der Personen

### **HRESULT GetUsers ([out, retval] [IOrgObjects](#) \*\* *pList*)**

**Beschreibung:** Diese Methode liefert eine Liste aller Personen-Objekte. Diese Methode entspricht dem Aufruf: 'GetOrgObjectByClassName("Person")'.

**Rückgabewerte:**

*IOrgObjects*: Liste aller Personen

### **HRESULT GetRoles ([out, retval] [IOrgObjects](#) \*\* *pList*)**

**Beschreibung:** Diese Methode liefert alle Rollen-Objekte. Diese Methode entspricht dem Aufruf: 'GetOrgObjectByClassName("Rolle")'.

**Rückgabewerte:**

*IOrgObjects*: Liste aller Rollen

**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

**HRESULT GetOrgObjectsByClassId([in] BSTR bstrClassId, [out,retval] IOrgObjects\*\* pIOrgObjects)**

**Beschreibung:** Diese Methode liefert Organisationsobjekte anhand der Id einer Organisationsklasse.

**Parameter:**

*BSTR*: Id der Klasse

**Rückgabewerte:**

*IOrgClasses*: Liste von Objekten der angegebenen Klasse.

**HRESULT GetOrganisation([in] BSTR ProcessID, [out,retval] IOrganisation\*\* pIOrganisation)**

**Beschreibung:** Diese Methode liefert ein Organisationsobjekt, das um die Rollen (Prozessrollen, virtuelle Rollen) des angegebenen Prozesses angereichert ist. Achtung, dieses Objekt ist nur im Kontext des Prozesses zu verwenden.

**Parameter:**

*BSTR*: Id des Prozesses

**Rückgabewerte:**

*IOrganisation*: Liste von Objekten der angegebenen Klasse.

**HRESULT GetProcessRoles([out,retval] IOrgObjects\*\* pIOrgObjects)**

**Beschreibung:** Diese Methode liefert alle Prozessrollen und virtuellen Rollen.

Achtung: Diese Methode kann nur auf Organisationsobjekten aufgerufen werden, die über [GetOrganisation](#)([in] BSTR ProcessID, [out,retval] IOrganisation\*\* pIOrganisation) initialisiert

wurden, da nur dann ein Bezug zu einem Prozess existiert. Andere Aufrufe führen zu einer Fehlermeldung.

**Rückgabewerte:**

*IOrgObjects*: Liste von Prozessrollen und virtuellen Rollen.



## IOrgAttribute Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt ein Attribut eines Organisationsobjektes.

### Öffentliche Methoden

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

### Dokumentation der Elementfunktionen

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

**HRESULT [Value](#) ([out, retval] BSTR \* pVal)**  
**Beschreibung:** Diese Methode liefert den Wert des Attributes.  
**Rückgabewerte:**  
*BSTR: Wert*

**HRESULT [Id](#) ([out, retval] BSTR \* pVal)**  
**Beschreibung:** Diese Methode liefert den Namen eines Attributes.  
**Rückgabewerte:**  
*BSTR: Name*

## IOrgAttributes Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine statische Liste aller Attribute eines Organisationsobjekts. Die 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

### Ausführliche Beschreibung

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](#)

### 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 Organisationsklasse

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

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

**Rückgabewerte:**

*BSTR*: Name der Organisationsklasse

### **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. Zugriff auf diese 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)

Set ListUser = wforganisation.GetUsers

Set ListeRoles = wforganisation.GetRoles
```

Siehe auch: [IOrganisation](#)

### Öffentliche Methoden

```
HRESULT Attributes ([out, retval] IOrgAttributes **pVal)
HRESULT GetPredecessorsByClassName ([in] BSTR bstrClassName,[in]
VARIANT_BOOL bDirectOnly,[out, retval] IOrgObjects **pList)
HRESULT GetPredecessorsByName ([in] BSTR bstrName,[out, retval]
IOrgObjects **pList)
HRESULT GetSuccessorsByClassName ([in]BSTR bstrClassId,[out, retval]
IOrgObjects **SuccList)
HRESULT GetSuccessorsByName ([in]BSTR bstrName,[out, retval] IOrgObjects
**pList)
HRESULT Id ([out, retval] BSTR *pVal)
HRESULT Name ([out, retval] BSTR *pVal)
HRESULT Predecessors ([out, retval] IOrgObjects **pVal)
HRESULT Substitutes ([out, retval]IOrgObjects **pVal)
HRESULT Successors ([out, retval] IOrgObjects **pVal)
```

## Dokumentation der Elementfunktionen

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

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

**Rückgabewerte:**

*BSTR*: ObjektId

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

**Beschreibung:** Diese Methode liefert den Objektnamen.

**Rückgabewerte:**

*BSTR*: Objektname

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

**Beschreibung:** Diese Methode liefert eine Liste aller direkten Vorgängerobjekte aus der Organisationsstruktur.

**Rückgabewerte:**

*IOrgObjects*: Liste aller Vorgängerobjekte

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

**Beschreibung:** Diese Methode liefert eine Liste aller direkten Nachfolgerobjekte aus der Organisationsstruktur.

**Rückgabewerte:**

*IOrgObjects:* Liste aller Nachfolgerobjekte

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

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

**Rückgabewerte:**

*IOrgObjAttributes:* Liste der Organisationsobjektattribute

### **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

### **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



**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

**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

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

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

**Rückgabewerte:**

*IOrgObjects*: Liste aller Stellvertreter

## 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. Kann auch eine Vorlage für eine Laufliste repräsentieren. Diese Schnittstelle ist von IComCollection abgeleitet und enthält Elemente vom Typ IRoutingListEntry.

Änderungen an der Laufliste werden nur durch das 'BeforeOpen Event' wirksam.

**Zugriff auf die Liste:** siehe IActiveWorkItem

Siehe auch: [IRoutingListEntry](#), [IRoutingListItem](#)

#### Öffentliche Methoden

**HRESULT** [AdhocModActivityId](#)([out, retval] BSTR\* pVal)  
**HRESULT** [CreateNewEntry](#)([out, retval] [IRoutingListEntry](#)\*\* pEntry)  
**HRESULT** [CreateNewItem](#)([in] IWFPProcessActivity, [out, retval] [IRoutingListItem](#)\*\* pItem)  
**HRESULT** [LegalItemActivities](#)([out, retval] IWFPProcessActivities\*\* pVal)  
**HRESULT** [ID](#)([out, retval] BSTR\* pVal)  
**HRESULT** [ModWorkitemId](#)([out, retval] BSTR\* pVal)  
**HRESULT** [ProcessId](#)([out, retval] BSTR\* pVal)  
**HRESULT** [WorkitemId](#)([out, retval] BSTR\* pVal)  
**HRESULT** [WorkflowId](#)([out, retval] BSTR\* pVal)  
**HRESULT** [WorkflowVersion](#)([out, retval] long\* 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** [IsIgnoreFlags](#)([out, retval] VARIANT\_BOOL\* pbVal)  
**HRESULT** [IsIgnoreFlags](#) ([in] VARIANT\_BOOL bVal)  
**HRESULT** [IsValid](#)([out, retval] VARIANT\_BOOL \* pbVal)

## Dokumentation der Elementfunktionen

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

**Beschreibung:** Diese Methode Id der Adhocaktivität im Modell.

**Rückgabewerte:**

*BSTR*: Id der Adhocaktivität im Modell

**Beispiel:**

```
Aid = workitem.RoutingList.AdhocModActivityId
```

### **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 ihn 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 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.LegalItemActivities
```

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

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

**Rückgabewerte:**

*BSTR*: Id des Prozesses

**Beispiel:**

```
Pid = workitem.RoutingList.ProcessId
```

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

**Beschreibung:** Diese Methode liefert die Id des Arbeitsschrittes m Prozess.

**Rückgabewerte:**

*BSTR*: Id des Arbeitsschrittes im Prozess

**Beispiel:**

```
Wid = workitem.RoutingList.WorkitemId
```

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

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

**Rückgabewerte:**

*BSTR*: Id des Workflowmodells

**Beispiel:**

```
Wid = workitem.RoutingList.WorkflowId
```

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

**Beschreibung:** Diese Methode liefert die Version des Workflowmodells.

**Rückgabewerte:**

*long*: Version des Workflowmodells

**Beispiel:**

```
version = workitem.RoutingList.WorkflowVersion
```

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

**Beschreibung:** Diese Methode liefert die Id des Arbeitsschrittes im Modell.

**Rückgabewerte:**

*long*: Id des Arbeitsschrittes im Modell

**Beispiel:**

```
MWId = workitem.RoutingList.ModWorkitemId
```

## **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 IsIgnoreFlags([out, retval] VARIANT\_BOOL\* pbVal)**

**Beschreibung:** Diese Methode gibt an, ob die Flags IsExpandable, IsChangeable, IsDeleteable dieser Liste, ihrer Entries und Items ignoriert werden sollen.

**Rückgabewerte:**

VARIANT\_BOOL: true, wenn die Flags IsExpandable, IsChangeable, IsDeleteable dieser Liste, ihrer Entries und Items ignoriert werden sollen.

### **HRESULT IsIgnoreFlags ([in] VARIANT\_BOOL bVal)**

**Beschreibung:** Diese Methode setzt, ob die Flags IsExpandable, IsChangeable, IsDeleteable dieser Liste, ihrer Entries und Items ignoriert werden sollen.

**Parameter:**

*VARIANT\_BOOL*: true, wenn die Flags IsExpandable, IsChangeable, IsDeleteable dieser Liste, ihrer Entries und Items ignoriert werden sollen

### **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 [ICollection](#) 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 [ActivityId](#)**([out, retval] BSTR\* pVal)  
**HRESULT [Activity](#)** ([in] IWFPProcessActivity\* pActivity)  
**HRESULT [ActivityName](#)**([out, retval] BSTR\* pVal)  
**HRESULT [ActivityName](#)** ([in] BSTR newVal)  
**HRESULT [ID](#)**([out, retval] BSTR\* pVal)  
**HRESULT [Remark](#)** ([out, retval] BSTR\* pVal)  
**HRESULT [Remark](#)** ([in] BSTR newVal)  
**HRESULT [ModelActivityName](#)**([out, retval] BSTR\* pVal)  
**HRESULT [ReminderPeriodId](#)**([out, retval] BSTR\* pVal)  
**HRESULT [ReminderPeriod](#)** ([in] [IAdhocReminderPeriodTemplate](#) period)  
**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 [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 Activity ([in] IWFPProcessActivity\* pActivity)**

**Beschreibung:** Diese Methode setzt die Aktivität dieses Elements der Abarbeitungsliste.

**Parameter:**

*IWFPProcessActivity*: Aktivität 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.

**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 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 newItem = workitem.RoutingList.CreateNewItem(LegalItemActivities.Item(0))
IID = newItem.ID
```

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

**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 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

### **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](#)

## 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 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
- -4 = Angabe des im Infofenster anzuzeigenden Objekts im ResultString in der Form: Objekt-ID, Objekt-Typ

### **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 setzt den Ergebnistext oder die Fehlermeldung.

**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](#)

## IWFAActivities Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Liste aller Vorgangsschritte eines Prozesses. Die Schnittstelle wird vom enaio® – Workflow Administrator verwendet. Zugriff auf diese Schnittstelle über:

- [IWFCClient::GetActivitiesFromProcess\(\)](#)

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



## IWFAActivity Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt einen Vorgangsschritt. Die Schnittstelle wird vom enaio® - Workflow Administrator verwendet. Zugriff auf diese Schnittstelle über:

- [IWFCClient::GetActivitiesFromProcess\(\)](#)

### Öffentliche Methoden

**HRESULT [AccessTime](#)** ([out, retval] long \*pIVal)  
**HRESULT [ClosureTime](#)** ([out, retval] long \*pIVal)  
**HRESULT [CreationTime](#)** ([out, retval] long \*pIVal)  
**HRESULT [EndTime](#)** ([out, retval] long \*pIVal)  
**HRESULT [ExecutionPoints](#)** ([out, retval] VARIANT \*pVal)  
**HRESULT [ID](#)** ([out, retval] BSTR \*pbstrVal)  
**HRESULT [IsWorkItem](#)** ([out, retval] long \*pIVal)  
**HRESULT [Name](#)** ([out, retval] BSTR \*pbstrVal)  
**HRESULT [OwnerID](#)** ([out, retval] BSTR \*pbstrVal)  
**HRESULT [OwnerName](#)** ([out, retval] BSTR \*pbstrVal)  
**HRESULT [Performers](#)** ([out, retval] [IWFAActivityPerformers](#) \*\*ppActivityPerformers)  
**HRESULT [PreSuspendState](#)** ([out, retval] long\* pIVal)  
**HRESULT [RelativID](#)** ([out, retval] BSTR \*pbstrVal)  
**HRESULT [ReminderState](#)** ([out, retval] long \*pIVal)  
**HRESULT [ReminderTime](#)** ([out, retval] long \*pIVal)  
**HRESULT [Resume](#)** ()  
**HRESULT [RLoopId](#)** ()  
**HRESULT [Rollback](#)** ([in] VARIANT\_BOOL bDoRollback, [in]long lExecPoint, [out, retval] [IWFAActivities](#)\*\* ppActivities)  
**HRESULT [State](#)** ([out, retval] long \*pIVal)  
**HRESULT [Suspend](#)** ()

### Dokumentation der Elementfunktionen

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

**Beschreibung:** Diese Methode liefert die Workflow-Id des Vorgangsschritts.

**Rückgabewerte:**

*BSTR*: Workflow-Id

**HRESULT RelativID** ([out, retval] BSTR \* *pbstrVal*)

**Beschreibung:** Diese Methode liefert die Instanz-Id des Vorgangsschritts.

**Rückgabewerte:**

*BSTR*: Instanz-Id

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

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

**Rückgabewerte:**

*BSTR*: Name

### **HRESULT OwnerID ([out, retval] BSTR \* *pbstrVal*)**

**Beschreibung:** Diese Methode liefert die Bearbeiter-Id zum Vorgangsschritt. Die Eigenschaft ist nur gesetzt, wenn der Vorgangsschritt personalisiert wurde.

**Rückgabewerte:**

*BSTR*: Bearbeiter-Id

### **HRESULT OwnerName ([out, retval] BSTR \* *pbstrVal*)**

**Beschreibung:** Diese Methode liefert die Bearbeiternamen zum Vorgangsschritt. Die Eigenschaft ist nur gesetzt, wenn der Vorgangsschritt personalisiert wurde.

**Rückgabewerte:**

*BSTR*: Bearbeitername

### **HRESULT State ([out, retval] long \* *pIVal*)**

**Beschreibung:** Diese Methode liefert den Status des Vorgangsschritts.

**Rückgabewerte:**

*long*: [Status](#)

### **HRESULT CreationTime ([out, retval] long \* *pIVal*)**

**Beschreibung:** Diese Methode gibt an, wann der Vorgangsschritt am Server erstellt wurde.

**Rückgabewerte:**

*long*: Erstellzeit in Form eines Zeitstempels

### **HRESULT AccessTime ([out, retval] long \* *pIVal*)**

**Beschreibung:** Diese Methode gibt an, wann der Vorgangsschritt personalisiert wurde.

**Rückgabewerte:**

*long*: Zeitpunkt der Personalisierung in Form eines Zeitstempels

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

**Beschreibung:** Diese Methode gibt an, wann der Vorgangsschritt beendet wurde.

**Rückgabewerte:**

*long*: Zeitpunkt der Personalisierung in Form eines Zeitstempels

### **HRESULT ClosureTime ([out, retval] long \* pIVal)**

**Beschreibung:** Diese Methode liefert die Sperrfrist zum Vorgangsschritt.

**Rückgabewerte:**

*long*: Sperrfrist in Form eines Zeitstempels

### **HRESULT ReminderTime ([out, retval] long \* pIVal)**

**Beschreibung:** Diese Methode liefert die Mahnfrist zum Vorgangsschritt.

**Rückgabewerte:**

*long*: Mahnfrist in Form eines Zeitstempels

### **HRESULT ReminderState ([out, retval] long \* pIVal)**

**Beschreibung:** Diese Methode gibt an, ob die Mahnfrist überschritten wurde.

**Rückgabewerte:**

*long*: 1 = Mahnfrist überschritten, ansonsten 0

### **HRESULT Performers ([out, retval] [IWFAActivityPerformers](#) \*\* ppActivityPerformers)**

**Beschreibung:** Diese Methode liefert eine Liste aller Teilnehmer des Vorgangsschritts.

**Rückgabewerte:**

*IWFAActivityPerformers*: Liste aller Teilnehmer

### **HRESULT IsWorkItem ([out, retval] long \* pIVal)**

**Beschreibung:** Diese Methode gibt an, ob der Vorgangsschritt im Eingangskorb erscheint und somit von einem Benutzer bearbeitet werden kann.

**Rückgabewerte:**

*long*: 1 = Vorgang kann bearbeitet werden, ansonsten 0

### **HRESULT Rollback([in] VARIANT\_BOOL bDoRollback, [in]long IExecPoint, [out, retval] [IWFAActivities](#)\*\* ppActivities)**

**Beschreibung:** Diese Methode setzt einen Prozess auf die angegebene Aktivität zurück und liefert alle Aktivitäten des Prozesses, die dadurch gelöscht werden.

**Parameter:**

*VARIANT\_BOOL*: FALSE = Zurücksetzen wird nicht durchgeführt; TRUE = Prozess wird zurückgesetzt

*LONG*: Startpunkt an der Aktivität

100 = Aktivität ist erstellt

110 = nach Ausführung des StartActivity-Events

190 = vor Ausführung des EndActivity-Events

200 = Aktivität ist beendet

#### Rückgabewerte:

*IWFActivities*: Liste aller Aktivitäten die nach dem Zurücksetzen gelöscht wurden (DoRollback = TRUE) bzw. würden (DoRollback = FALSE)

Siehe auch: [ExecutionPoints](#)

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

**Beschreibung:** Diese Methode liefert den Status des Vorgangsschritts, bevor dieser angehalten wurde.

#### Rückgabewerte:

*LONG*: Status des Vorgangsschritts, bevor dieser angehalten wurde (siehe [Aktivitätsstatus](#))

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

**Beschreibung:** Diese Methode liefert alle Startpunkte an einer Aktivität. Die Startpunkte werden in der Methode Rollback verwendet

#### Rückgabewerte:

*VARIANT*: Kommaseparierte Startpunkte

Siehe auch: [Rollback](#)

### **HRESULT Resume ()**

**Beschreibung:** Diese Methode gibt diesen Vorgangsschritt für die Bearbeitung wieder frei.

Siehe auch: [Suspend](#)

### **HRESULT Suspend ()**

**Beschreibung:** Diese Methode sperrt diesen Vorgangsschritt für die Bearbeitung.

Siehe auch: [Resume](#)

### **HRESULT RLoopId ([out, retval] BSTR \* pbstrVal)**

**Beschreibung:** Diese Methode liefert die InstanzId der umgebenen Schleife, wenn es diese gibt.

#### Rückgabewerte:

*BSTR*: InstanzId der umgebenenSchleife

## IWFActivities Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Liste aller laufenden Aktivitäten. Die Liste wird im enaio® - Workflow Client für die Registerkarte "Laufende Vorgänge" verwendet. Zugriff auf diese Schnittstelle über:

- [IWFCClient::RunningActivities\(\)](#)

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

## IWFActivity Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine laufende Aktivität. Die Schnittstelle wird im enaio® - Workflow Client für die Registerkarte "Laufende Vorgänge" verwendet. Zugriff auf diese Schnittstelle über:

- [IWFCClient::RunningActivities\(\)](#)

### Öffentliche Methoden

**HRESULT [ClosureTime](#)** ([out, retval] long \*pVal)  
**HRESULT [GetVarName](#)** ([in] long IVarNum,[out, retval]BSTR \*pbstrVarName)  
**HRESULT [GetVarValue](#)** ([in] long IVarNum,[out, retval] BSTR \*pbstrVarValue)  
**HRESULT [GetVarDataType](#)**([in] long IVarNum,[out, retval] VarDataType \*pVal)  
**HRESULT [ID](#)** ([out, retval] BSTR \*pVal)  
**HRESULT [Name](#)** ([out, retval] BSTR \*pVal)  
**HRESULT [OverTime](#)** ([out, retval] VARIANT\_BOOL \*pVal)  
**HRESULT [ProcessCreationTime](#)** ([out, retval] long \*pVal)  
**HRESULT [ProcessID](#)** ([out, retval] BSTR \*pVal)  
**HRESULT [ProcessName](#)** ([out, retval] BSTR \*pVal)  
**HRESULT [ReminderTime](#)** ([out, retval] long \*pVal)  
**HRESULT [State](#)** ([out, retval] long \*pVal)  
**HRESULT [UserName](#)** ([out, retval] BSTR \*pVal)  
**HRESULT [VarCount](#)** ([out, retval] long \*pVal)  
**HRESULT [ObjectId](#)** ([out, retval] long \*pVal)

### Dokumentation der Elementfunktionen

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

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

**Rückgabewerte:**

*BSTR*: Name der Aktivität

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

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

**Rückgabewerte:**

*BSTR*: Id der Aktivität

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

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

**Rückgabewerte:**

*BSTR*: Name des Prozesses

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

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

**Rückgabewerte:**

*BSTR*: Prozessid

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

**Beschreibung:** Diese Methode gibt an, wann der Prozess gestartet wurde. Die Startzeit wird in Form eines Zeitstempels zurückgegeben.

**Rückgabewerte:**

*long*: Startzeit

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

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

**Rückgabewerte:**

*BSTR*: Bearbeiter der Aktivität

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

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

**Rückgabewerte:**

*long*: [Status](#)

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

**Beschreibung:** Diese Methode gibt an, ob die Mahnfrist überschritten wurde.

**Rückgabewerte:**

*VARIANT\_BOOL*: TRUE = Mahnfrist überschritten

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

**Beschreibung:** Diese Methode liefert Sperrfrist in Form eines Zeitstempels. Sperrfrist: ab diesem Zeitpunkt kann der Vorgangsschritt erst bearbeitet werden.

**Rückgabewerte:**



*long*: Sperrfrist

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

**Beschreibung:** Diese Methode liefert Mahnfrist in Form eines Zeitstempels. Mahnfrist: bis zu diesem Zeitpunkt muss der Vorgangsschritt bearbeitet sein.

**Rückgabewerte:**

*long*: Mahnfrist

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

**Beschreibung:** Diese Methode liefert die Anzahl der Workflowvariablen die unter 'laufende Vorgänge' angezeigt werden sollen.

**Rückgabewerte:**

*long*: Anzahl der Workflowvariablen

Siehe auch:

[GetVarName\(\)](#), [GetVarValue\(\)](#)

### **HRESULT GetVarName ([in] long lVarNum, [out, retval] BSTR \* pbstrVarName)**

**Beschreibung:** Diese Methode liefert den Namen einer Workflowvariablen die unter 'laufende Vorgänge' angezeigt werden sollen.

**Parameter:**

*long*: Index der geforderten Variablen.

**Rückgabewerte:**

*BSTR*: Name der Variable

Siehe auch:

[GetVarValue\(\)](#), [VarCount\(\)](#)

### **HRESULT GetVarValue ([in] long *IVarNum*, [out, retval] BSTR \* *pbstrVarValue*)**

**Beschreibung:** Diese Methode liefert den Wert einer Workflowvariablen die unter 'laufende Vorgänge' angezeigt werden sollen.

**Parameter:**

*long*: Index der Workflowvariablen

**Rückgabewerte:**

*BSTR*: Wert der Variable

**Siehe auch:**

[GetVarName\(\)](#), [VarCount\(\)](#)

### **HRESULT GetVarDataType ([in] long *IVarNum*, [out, retval] VarDataType \**pVal*)**

**Beschreibung:** Diese Methode liefert den Datentypen einer Workflowvariablen die unter 'laufende Vorgänge' angezeigt werden sollen.

**Parameter:**

*long*: Index der Workflowvariablen

**Rückgabewerte:**

*VarDataType*: Datentyp der Variable

**Siehe auch:**

[GetVarName\(\)](#), [VarCount\(\)](#) , [IWFVariable::VarDataType\(\)](#)

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

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

**Rückgabewerte:**

*long*: Objektid

## IWFActivityPerformer Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt einen Teilnehmer einer Aktivität.

### Öffentliche Methoden

**HRESULT [Absent](#)** ([out, retval] VARIANT\_BOOL \*pVal)  
**HRESULT [ObjectId](#)** ([out, retval] BSTR \*pVal)  
**HRESULT [Substitute](#)** ([out, retval] VARIANT\_BOOL \*pVal)

### Dokumentation der Elementfunktionen

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

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

**Rückgabewerte:**

*BSTR*: Objektid

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

**Beschreibung:** Diese Methode gibt an, ob der Teilnehmer bzw. die Rolle abwesend ist. Eine Rollen gilt als abwesend, wenn alle Personen, die der Rolle zugeordnet sind, abwesend sind.

**Rückgabewerte:**

*VARIANT\_BOOL*: True = abwesend, False = anwesend

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

**Beschreibung:** Diese Methode gibt an, ob der Teilnehmer die Aktivität in Vertretung bearbeitet.

**Rückgabewerte:**

*VARIANT\_BOOL*: True = in Vertretung, ansonsten False

## IWFActivityPerformers Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Liste aller Teilnehmer einer Aktivität. Zugriff auf diese Schnittstelle über:

- [IWFCClient::GetActivityPerformers](#) ()
- [IWFAActivity::Performers](#) ()

Elemente der Liste: siehe [IWFActivityPerformer](#)

## IWFHistActivities Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Liste von Vorgangsschritten für Historienverwaltung. Die Schnittstelle wird vom enaio® - Workflow Administrator verwendet. Zugriff auf diese Schnittstelle über:

- [IWFCClient::GetHistActivitiesFromProcess\(\)](#)

Elemente der Liste: [IWFHistActivity](#)

## IWFAHistActivity Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt einen Vorgangsschritt für die Historienverwaltung. Die Schnittstelle wird vom enaio® - Workflow Administrator verwendet. Zugriff auf diese Schnittstelle über:

- [IWFCClient::GetHistActivitiesFromProcess\(\)](#)

### Öffentliche Methoden

**HRESULT [EntryNr](#) ([out, retval] long \*pVal)**  
**HRESULT [Name](#) ([out, retval] BSTR \*pVal)**  
**HRESULT [RActivityID](#) ([out, retval] BSTR \*pVal)**

### Dokumentation der Elementfunktionen

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

**Beschreibung:** Diese Methode liefert die Instanz-Id des Vorgangsschritts.

**Rückgabewerte:**

*BSTR*: Instanz-Id

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

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

**Rückgabewerte:**

*BSTR*: Name

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

**Beschreibung:** Diese Methode liefert die Nummer des Historieneintrags. Die Eintragsnummern spiegeln den zeitlichen Ablauf der Einträge wieder.

**Rückgabewerte:**

*BSTR*: Nummer des Historieneintrags

## IWFHistEntries Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Liste von historischen Einträgen der Historienverwaltung. Die Schnittstelle wird vom enaio® - Workflow Administrator verwendet. Zugriff auf diese Schnittstelle über:

- [IWFClient::GetHistEntriesFromProcess\(\)](#)
- [IWFClient::GetHistEntriesFromRActivity\(\)](#)
- [IWFClient::GetHistEntriesFromTimer\(\)](#)

Elemente der Liste: [IWFHistEntry](#)

## IWFAHistEntry Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt einen historischen Eintrag der Historienverwaltung. Die Schnittstelle wird vom enaio® - Workflow Administrator verwendet. Zugriff auf diese Schnittstelle über:

- [IWFCClient::GetHistEntriesFromProcess\(\)](#)
- [IWFCClient::GetHistEntriesFromRActivity\(\)](#)
- [IWFCClient::GetHistEntriesFromTimer\(\)](#)

### Öffentliche Methoden

**HRESULT [EntryNr](#) ([out, retval] long \*pVal)**  
**HRESULT [HistID](#) ([out, retval] BSTR \*pVal)**  
**HRESULT [HistType](#) ([out, retval] long \*pVal)**  
**HRESULT [OrgID](#) ([out, retval] BSTR \*pVal)**  
**HRESULT [ProcessID](#) ([out, retval] BSTR \*pVal)**  
**HRESULT [RActivityID](#) ([out, retval] BSTR \*pVal)**  
**HRESULT [ServerID](#) ([out, retval] BSTR \*pVal)**  
**HRESULT [ServerName](#) ([out, retval] BSTR \*pVal)**  
**HRESULT [Time](#) ([out, retval] long \*pVal)**  
**HRESULT [UserID](#) ([out, retval] BSTR \*pVal)**  
**HRESULT [UserName](#) ([out, retval] BSTR \*pVal)**

### Dokumentation der Elementfunktionen

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

**Beschreibung:** Diese Methode liefert die Id des historischen Eintrags.

**Rückgabewerte:**

*BSTR*: Prozess-Id

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

**Beschreibung:** Diese Methode liefert die Prozess-Id.

**Rückgabewerte:**

*BSTR*: Prozess-Id

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

**Beschreibung:** Diese Methode liefert die Instanz-Id des Vorgangsschritts.

**Rückgabewerte:**

*BSTR*: Instanz-Id



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

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

**Rückgabewerte:**

*long*: Eintragstyp

**Hinweis:** mögliche Werte

- 1 = PREPAREPROCESS
- 2 = STARTPROCESS
- 3 = ENDPROCESS
- 4 = PREPAREACTIVITY
- 5 = STARTACTIVITY
- 6 = ENDACTIVITY
- 7 = COPYACTIVITYVARIABLES
- 8 = HALTACTIVITY
- 9 = REACTIVATEACTIVITY
- 10 = STARTWORKITEM
- 11 = PERSONALIZED
- 12 = DEPERSONALIZED
- 13 = SAVEWORKITEM
- 14 = ENDWORKITEM
- 15 = CREATETIMER
- 16 = CANCELTIMER
- 17 = REMIND
- 18 = DELAYED
- 19 = STARTSCRIPT
- 20 = ENDSRIPT
- 21 = SYSSUSPEND
- 22 = SETACTIVITYPERFORMER
- 23 = DELETEACTIVITYPERFORMER

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

**Beschreibung:** Diese Methode liefert den Zeitpunkt der Eintragserstellung.

**Rückgabewerte:**

*BSTR*: Organisations-Id

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

**Beschreibung:** Diese Methode liefert die Organisations-Id.

**Rückgabewerte:**

*BSTR*: Organisations-Id

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

**Beschreibung:** Diese Methode liefert die Benutzer-Id.

**Rückgabewerte:**

*BSTR*: Benutzer-Id

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

**Beschreibung:** Diese Methode liefert den Benutzernamen.

**Rückgabewerte:**

*BSTR*: Benutzername

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

**Beschreibung:** Diese Methode liefert die Server-Id.

**Rückgabewerte:**

*BSTR*: Server-Id

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

**Beschreibung:** Diese Methode liefert den Namen des Server, auf dem die Aktivität erstellt wurde.

**Rückgabewerte:**

*BSTR*: Servername

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

**Beschreibung:** Diese Methode liefert die Nummer des historischen Eintrags. Die Eintragsnummern spiegeln den zeitlichen Ablauf (1 = erste Aktion n = letzte Aktion) wieder.

**Rückgabewerte:**

*BSTR*: Eintragsnummer

## IWFAHistTimer Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine historische Mahn-/Sperrfrist der Historienverwaltung. Die Schnittstelle wird vom enaio® - Workflow Administrator verwendet. Zugriff auf diese Schnittstelle über:

- [IWFCClient::GetHistTimersFromProcess\(\)](#)

### Öffentliche Methoden

**HRESULT [DestinationTime](#)** ([out, retval] long \*pVal)  
**HRESULT [DestinationType](#)** ([out, retval] long \*pVal)  
**HRESULT [FromActivityID](#)** ([out, retval] BSTR \*pVal)  
**HRESULT [FromActivityName](#)** ([out, retval] BSTR \*pVal)  
**HRESULT [ProcessID](#)** ([out, retval] BSTR \*pVal)  
**HRESULT [TimerID](#)** ([out, retval] BSTR \*pVal)  
**HRESULT [TimerType](#)** ([out, retval] long \*pVal)  
**HRESULT [ToActivityID](#)** ([out, retval] BSTR \*pVal)  
**HRESULT [ToActivityName](#)** ([out, retval] BSTR \*pVal)

### Dokumentation der Elementfunktionen

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

**Beschreibung:** Diese Methode liefert die Id der historischen Mahn-/Sperrfrist.

**Rückgabewerte:**

*BSTR*: Id

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

**Beschreibung:** Diese Methode liefert die Id des Prozesses, dem die historischen Mahn-/Sperrfrist zugeordnet war.

**Rückgabewerte:**

*BSTR*: Prozess-Id

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

**Beschreibung:** Diese Methode liefert die Id der Aktivität, die als Anfangsaktivität für die Frist definiert wurde.

**Rückgabewerte:**

*BSTR*: Id der Aktivität

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

**Beschreibung:** Diese Methode liefert den Namen der Aktivität, die als Anfangsaktivität für die Frist definiert wurde.

**Rückgabewerte:**

*BSTR*: Name der Aktivität

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

**Beschreibung:** Diese Methode liefert die Id der Aktivität, die als Endaktivität für die Frist definiert wurde.

**Rückgabewerte:**

*BSTR*: Id der Aktivität

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

**Beschreibung:** Diese Methode liefert den Namen der Aktivität, die als Endaktivität für die Frist definiert wurde.

**Rückgabewerte:**

*BSTR*: Name der Aktivität

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

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

**Rückgabewerte:**

*long*: Typ der Frist

**Hinweis:** mögliche Werte

- 0 = Sperrfrist
- 1 = Mahnfrist

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

**Beschreibung:** Diese Methode liefert die Endzeitpunkt der Frist in Form eines Zeitstempels.

**Rückgabewerte:**

*long*: Endzeitpunkt der Frist

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

**Beschreibung:** Diese Methode gibt an, ob die Frist am Anfang bzw. Ende der Aktivität gültig ist.

**Rückgabewerte:**

*long*: Typ

**Hinweis:** mögliche Werte

- 0 = Anfang der Aktivität
- 1 = Ende der Aktivität

## IWFHistTimers Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Liste von historischen Mahn-/Sperrfristen der Historienverwaltung. Die Schnittstelle wird vom enaio® - Workflow Administrator verwendet. Zugriff auf diese Schnittstelle über:

- [IWFCClient::GetHistTimersFromProcess\(\)](#)

Elemente der Liste: [IWFHistTimer](#)

## IWFAHistWorkflow Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt ein Workflowmodell für Historienverwaltung. Die Schnittstelle wird vom enaio® – Workflow Administrator verwendet. Zugriff auf diese Schnittstelle über:

- [IWFCClient::HistWorkflows\(\)](#)

### Öffentliche Methoden

**HRESULT [HistWFID](#) ([out, retval] BSTR \*pVal)**  
**HRESULT [ProcessCount](#) ([out, retval] long \*pVal)**  
**HRESULT [WFID](#) ([out, retval] BSTR \*pVal)**  
**HRESULT [WFName](#) ([out, retval] BSTR \*pVal)**  
**HRESULT [WFVersion](#) ([out, retval] long \*pVal)**  
**HRESULT [WFFamilyId](#) ([out, retval] BSTR \*pVal)**  
**HRESULT [WFFamilyName](#) ([out, retval] BSTR \*pVal)**

### Dokumentation der Elementfunktionen

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

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

**Rückgabewerte:**

*BSTR*: ID aus dem Workflowmodell

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

**Beschreibung:** Diese Methode liefert den Namen aus dem Workflowmodell.

**Rückgabewerte:**

*BSTR*: Name aus dem Workflowmodell

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

**Beschreibung:** Diese Methode gibt an, wie oft das Modell gestartet wurde.

**Rückgabewerte:**

*long*: Anzahl der Starts

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

**Beschreibung:** Diese Methode liefert die Versionsnummer des Workflowmodells.



**Rückgabewerte:**

*long*: Versionsnummer

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

**Beschreibung:** Diese Methode liefert die Historien-ID des Workflowmodell.

**Rückgabewerte:**

*BSTR*: Historien-ID

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

**Beschreibung:** Diese Methode liefert die Id der Workflowfamilie, in der sich das Workflowmodell befindet.

**Rückgabewerte:**

*BSTR*: Id der Workflowfamilie

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

**Beschreibung:** Diese Methode liefert den Namen der Workflowfamilie, in der sich das Workflowmodell befindet.

**Rückgabewerte:**

*BSTR*: Name der Workflowfamilie

## IWFHistWorkflows Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Liste von historischen Workflows. Die Schnittstelle wird vom enaio® - Workflow Administrator genutzt. Zugriff auf diese Schnittstelle über:

- [IWFCClient::HistWorkflows\(\)](#)

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

## IWFHistWorkItemRelEntries Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Liste von hist. Eingangskorbeinträgen. Die Liste wird vom enaio® - Workflow Administrator verwendet. Zugriff auf diese Schnittstelle über:

- [IWFClient::GetHistWorkItemRelEntriesByUser\(\)](#)
- [IWFClient::GetHistWorkItemRelEntriesByActivity\(\)](#)

Elemente der Liste: siehe [IWFHistWorkItemRelEntry](#)

## IWFHistWorkItemRelEntry Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt einen hist. Eingangskorbeintrag. Die Schnittstelle wird vom enaio® - Workflow Administrator verwendet. Zugriff auf diese Schnittstelle über:

- [IWFCClient::GetHistWorkItemRelEntriesByUser\(\)](#)
- [IWFCClient::GetHistWorkItemRelEntriesByActivity\(\)](#)

### Öffentliche Methoden

**HRESULT [ActivityName](#) ([out, retval] BSTR \*pVal)**  
**HRESULT [EntryNr](#) ([out, retval] long \*pVal)**  
**HRESULT [HistID](#) ([out, retval] BSTR \*pVal)**  
**HRESULT [HistType](#) ([out, retval] long \*pVal)**  
**HRESULT [OrgID](#) ([out, retval] BSTR \*pVal)**  
**HRESULT [RActivityID](#) ([out, retval] BSTR \*pVal)**  
**HRESULT [Reason](#) ([out, retval] long \*pVal)**  
**HRESULT [Time](#) ([out, retval] long \*pVal)**  
**HRESULT [UserID](#) ([out, retval] BSTR \*pVal)**  
**HRESULT [UserName](#) ([out, retval] BSTR \*pVal)**

### Dokumentation der Elementfunktionen

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

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

**Rückgabewerte:**

*BSTR*: Instanz-Id

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

**Beschreibung:** Diese Methode liefert den Namen der Aktivität. Nur verfügbar bei Aufruf von [IWFCClient::GetHistWorkItemRelEntriesByUser\(\)](#) .

**Rückgabewerte:**

*BSTR*: Name

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

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

**Rückgabewerte:**

*BSTR*: Id

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

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

**Rückgabewerte:**

*BSTR*: Name

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

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

**Rückgabewerte:**

*BSTR*: Name

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

**Beschreibung:** Diese Methode gibt an, warum dieser Vorgangsschritt im Eingangskorb war.

**Rückgabewerte:**

*long*: Grund

**Hinweis:** mögliche Werte

- 0 = als Bearbeiter zugewiesen
- 1 = Aufgrund einer Mahnfrist zugewiesen
- 2 = vom WFAdmin zugewiesen

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

**Beschreibung:** Diese Methode liefert die Id des historischen Eintrags.

**Rückgabewerte:**

*BSTR*: Id

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

**Beschreibung:** Diese Methode gibt den History-Typ des entsprechenden History-Eintrags an.

**Rückgabewerte:**

*long*: History-Typ

**Hinweis:** mögliche Werte

- 1 = PREPAREPROCESS
- 2 = STARTPROCESS
- 3 = ENDPROCESS
- 4 = PREPAREACTIVITY
- 5 = STARTACTIVITY
- 6 = ENDACTIVITY
- 7 = COPYACTIVITYVARIABLES
- 8 = HALTACTIVITY
- 9 = REACTIVATEACTIVITY
- 10 = STARTWORKITEM
- 11 = PERSONALIZED
- 12 = DEPERSONALIZED
- 13 = SAVEWORKITEM
- 14 = ENDWORKITEM
- 15 = CREATETIMER
- 16 = CANCELTIMER
- 17 = REMIND
- 18 = DELAYED
- 19 = STARTSCRIPT
- 20 = ENDSRIPT
- 21 = SYSSUSPEND
- 22 = SETACTIVITYPERFORMER
- 23 = DELETEACTIVITYPERFORMER

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

**Beschreibung:** Diese Methode liefert die Nummer historischen Eintrags. Die Eintragsnummer spiegelt den zeitlichen Ablauf wieder.

**Rückgabewerte:**

*long*: EntryNr

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

**Beschreibung:** Diese Methode gibt den Zeitpunkt des entsprechenden History-Eintrags an.

**Rückgabewerte:**

*long*: Time

## IWFAModel Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt ein Workflowmodell. Die Schnittstelle wird vom enaio® - Workflow Administrator verwendet. Zugriff auf diese Schnittstelle über:

- [IWFCClient::WFAModels\(\)](#)

### Öffentliche Methoden

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

**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 des Workflowmodells.

**Rückgabewerte:**

*BSTR*: ID des Workflowmodells

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

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

**Rückgabewerte:**

*BSTR*: Name des Workflowmodells

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

**Beschreibung:** Diese Methode liefert die Anzahl der laufenden Prozesse des Workflowmodells.

**Rückgabewerte:**

*long*: Anzahl der laufenden Prozesse



## IWFAModels Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Liste von Workflowmodellen. Die Schnittstelle wird vom enaio® - Workflow Administrator genutzt. Zugriff auf diese Schnittstelle über:

- [IWFCClient::WFAModels\(\)](#)

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

## IWFAProcess Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt - abhängig von der IWFClient-Methode die zur Erstellung des Schnittstellenobjektes verwendet wird - einen Prozess oder eine Aktivität im Eingangskorb. Die Schnittstelle wird vom enaio® - Workflow Administrator genutzt. Zugriff auf diese Schnittstelle erhält über:

- [IWFClient::GetProcessesFromModel\(\)](#)
- [IWFClient::GetProcessesFromUser\(\)](#)
- [IWFClient::GetProcessesFromRole\(\)](#)
- [IWFClient::GetHistProcessesFromModel\(\)](#)

Es werden durch diese Methoden nicht alle IWFAProcess-Eigenschaften gesetzt, in der Methodenbeschreibung finden Sie einen Hinweis welche davon betroffen sind.

### Öffentliche Methoden

**HRESULT [ActivityID](#)** ([out, retval] BSTR \*pVal)  
**HRESULT [ActivityName](#)** ([out, retval] BSTR \*pVal)  
**HRESULT [ActivityOwnerID](#)** ([out, retval] BSTR \*pVal)  
**HRESULT [ActivityOwnerName](#)** ([out, retval] BSTR \*pVal)  
**HRESULT [ActivityOwnerTime](#)** ([out, retval] long \*pVal)  
**HRESULT [ActivityReminderTime](#)** ([out, retval] long \*pVal)  
**HRESULT [ActivityState](#)** ([out, retval] long \*pVal)  
**HRESULT [ActivityTime](#)** ([out, retval] long \*pVal)  
**HRESULT [ActivityWorkItemCreationTime](#)** ([out, retval] long \*pVal)  
**HRESULT [CreationTime](#)** ([out, retval] long \*pVal)  
**HRESULT [CreatorID](#)** ([out, retval] BSTR \*pVal)  
**HRESULT [CreatorName](#)** ([out, retval] BSTR \*pVal)  
**HRESULT [DeleteProcess](#)** ([out]VARIANT \*pvarResult,[out]VARIANT \*pvarErrorString)  
**HRESULT [DePersonalize](#)**([in] BSTR bstrUserId, [out]VARIANT\* pvarResult, [out]VARIANT \*pvarErrorString)  
**HRESULT [EndTime](#)** ([out, retval] long \*pVal)  
**HRESULT [GetProcessProtocol](#)** ([out]VARIANT \*pbstrPath,[out]VARIANT \*pvarResult,[out]VARIANT \*pvarErrorString)  
**HRESULT [ID](#)** ([out, retval] BSTR \*pVal)  
**HRESULT [LastActivityID](#)** ([out, retval] BSTR \*pVal)  
**HRESULT [LastActivityName](#)** ([out, retval] BSTR \*pVal)  
**HRESULT [LastActivityTime](#)** ([out, retval] long \*pVal)  
**HRESULT [Name](#)** ([out, retval] BSTR \*pVal)  
**HRESULT [Personalize](#)**([in] BSTR bstrUserId, [in] BSTR bstrUserName, [out]VARIANT\* pvarResult, [out]VARIANT \*pvarErrorString)  
**HRESULT [PreSuspendState](#)** ([out, retval] long\* pVal)  
**HRESULT [ResumeProcess](#)** ([out]VARIANT\* pvarResult, [out]VARIANT \*pvarErrorString)  
**HRESULT [State](#)** ([out, retval] long \*pVal)  
**HRESULT [SuspendProcess](#)** ([out]VARIANT\* pvarResult, [out]VARIANT \*pvarErrorString)

## Dokumentation der Elementfunktionen

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

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

**Rückgabewerte:**

*BSTR*: Prozessid

**Hinweis:** verfügbar bei Aufruf von

- [IWFCClient::GetProcessesFromModel\(\)](#)
- [IWFCClient::GetProcessesFromRole\(\)](#)
- [IWFCClient::GetProcessesFromUser\(\)](#)
- [IWFCClient::GetHistProcessesFromModel\(\)](#)

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

**Beschreibung:** Diese Methode liefert den Prozessnamen. Der Prozessname setzt sich aus dem Instanznamen (aus dem Workflowmodell) und dem Zähler (wie oft wurde das Modell gestartet) zusammen.

**Rückgabewerte:**

*BSTR*: Prozessname

**Hinweis:** verfügbar bei Aufruf von

- [IWFCClient::GetProcessesFromModel\(\)](#)
- [IWFCClient::GetProcessesFromRole\(\)](#)
- [IWFCClient::GetProcessesFromUser\(\)](#)
- [IWFCClient::GetHistProcessesFromModel\(\)](#)

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

**Beschreibung:** Diese Methode liefert den Prozessstatus.

**Rückgabewerte:**

*BSTR:* [Prozessstatus](#)

**Hinweis:** verfügbar bei Aufruf von

- [IWFCClient::GetProcessesFromModel\(\)](#)
- [IWFCClient::GetProcessesFromRole\(\)](#)
- [IWFCClient::GetProcessesFromUser\(\)](#)
- [IWFCClient::GetHistProcessesFromModel\(\)](#)

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

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

**Rückgabewerte:**

*BSTR:* Id des Prozesserstellers

**Hinweis:** verfügbar bei Aufruf von

- [IWFCClient::GetProcessesFromModel\(\)](#)
- [IWFCClient::GetProcessesFromRole\(\)](#)
- [IWFCClient::GetProcessesFromUser\(\)](#)

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

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

**Rückgabewerte:**

*BSTR:* Name des Prozesserstellers

**Hinweis:** verfügbar bei Aufruf von

- [IWFCClient::GetProcessesFromModel\(\)](#)
- [IWFCClient::GetProcessesFromRole\(\)](#)
- [IWFCClient::GetProcessesFromUser\(\)](#)
- [IWFCClient::GetHistProcessesFromModel\(\)](#)

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

**Beschreibung:** Diese Methode liefert den Zeitpunkt der Prozesserstellung in Form eines Zeitstempels.

**Rückgabewerte:**

*long*: Zeitpunkt der Prozesserstellung

**Hinweis:** verfügbar bei Aufruf von

- [IWFClient::GetProcessesFromModel\(\)](#)
- [IWFClient::GetProcessesFromRole\(\)](#)
- [IWFClient::GetProcessesFromUser\(\)](#)
- [IWFClient::GetHistProcessesFromModel\(\)](#)

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

**Beschreibung:** Diese Methode liefert die Instanzid der Aktivität, die sich im Eingangskorb befindet.

**Rückgabewerte:**

*BSTR*: Instanzid

**Hinweis:** verfügbar bei Aufruf von

- [IWFClient::GetProcessesFromRole\(\)](#)
- [IWFClient::GetProcessesFromUser\(\)](#)

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

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

**Rückgabewerte:**

*BSTR*: Name der aktuellen Aktivität

**Hinweis:** verfügbar bei Aufruf von

- [IWFClient::GetProcessesFromRole\(\)](#)
- [IWFClient::GetProcessesFromUser\(\)](#)

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

**Beschreibung:** Diese Methode liefert die Erstellzeit der Aktivität am Server.

**Rückgabewerte:**

*long*: Erstellzeit in Form eines Zeitstempels

**Hinweis:** verfügbar bei Aufruf von

- [IWFCClient::GetProcessesFromRole\(\)](#)
- [IWFCClient::GetProcessesFromUser\(\)](#)

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

**Beschreibung:** Diese Methode liefert den Namen des Bearbeiters der aktuellen Aktivität. Die Eigenschaft nur gesetzt, wenn die Aktivität auch personalisiert ist.

**Rückgabewerte:**

*BSTR*: Name des Bearbeiters

**Hinweis:** verfügbar bei Aufruf von

- [IWFCClient::GetProcessesFromRole\(\)](#)

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

**Beschreibung:** Diese Methode liefert die Id des Bearbeiters der aktuellen Aktivität. Die Eigenschaft nur gesetzt, wenn die Aktivität auch personalisiert ist.

**Rückgabewerte:**

*BSTR*: Id des Bearbeiters

**Hinweis:** verfügbar bei Aufruf von

- [IWFCClient::GetProcessesFromRole\(\)](#)

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

**Beschreibung:** Diese Methode liefert den Zeitpunkt der Personalisierung der aktuellen Aktivität.

**Rückgabewerte:**

*long*: Zeitpunkt der Personalisierung in Form eines Zeitstempels

**Hinweis:** verfügbar bei Aufruf von

- [IWFCClient::GetProcessesFromRole\(\)](#)
- [IWFCClient::GetProcessesFromUser\(\)](#)

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

**Beschreibung:** Diese Methode liefert die Mahnfrist zur aktuellen Aktivität. Wenn keine Mahnfrist existiert wird der Wert 0 zurückgeliefert.

**Rückgabewerte:**

*long*: Mahnfrist in Form eines Zeitstempels

**Hinweis:** verfügbar bei Aufruf von

- [IWFCClient::GetProcessesFromRole\(\)](#)
- [IWFCClient::GetProcessesFromUser\(\)](#)

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

**Beschreibung:** Diese Methode liefert die Erstellzeit der Aktivität im Eingangskorb.

**Rückgabewerte:**

*long*: Erstellzeit in Form eines Zeitstempels

**Hinweis:** verfügbar bei Aufruf von

- [IWFCClient::GetProcessesFromRole\(\)](#)
- [IWFCClient::GetProcessesFromUser\(\)](#)

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

**Beschreibung:** Diese Methode liefert die Id aus dem Workflow-Modell der letzten Aktivität, die vor der aktuellen Aktivität ausgeführt wurde.

**Rückgabewerte:**

*BSTR*: Id aus dem Workflow-Modell

**Hinweis:** verfügbar bei Aufruf von

- [IWFCClient::GetProcessesFromModel\(\)](#)

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

**Beschreibung:** Diese Methode liefert den Namen der letzten Aktivität, die vor der aktuellen Aktivität ausgeführt wurde.

**Rückgabewerte:**

*BSTR*: Id aus dem Workflow-Modell

**Hinweis:** verfügbar bei Aufruf von

- [IWFCClient::GetProcessesFromModel\(\)](#)

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

**Beschreibung:** Diese Methode liefert den Bearbeitungszeitpunkt der letzten Aktivität, die vor der aktuellen Aktivität ausgeführt wurde.

**Rückgabewerte:**

*long*: Zeitpunkt der Bearbeitung

**Hinweis:** verfügbar bei Aufruf von

- [IWFCClient::GetProcessesFromModel\(\)](#)



### **HRESULT DeleteProcess ([out] VARIANT \* *pvarResult*, [out] VARIANT \* *pvarErrorString*)**

**Beschreibung:** Diese Methode löscht den laufenden Prozess.

**Parameter:**

*VARIANT*: hier wird der Fehlercode zurückgeliefert, 0 -> kein Fehler aufgetreten

*VARIANT*: hier wird ggf. der Fehlertext zurückgeliefert

**Hinweis:** verfügbar bei Aufruf von

- [IWFCClient::GetProcessesFromModel\(\)](#)
- [IWFCClient::GetProcessesFromRole\(\)](#)
- [IWFCClient::GetProcessesFromUser\(\)](#)

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

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

**Rückgabewerte:**

*long*: Status der aktuellen Aktivität (siehe [Status](#))

**Hinweis:** verfügbar bei Aufruf von

- [IWFCClient::GetProcessesFromRole\(\)](#)
- [IWFCClient::GetProcessesFromUser\(\)](#)

### **HRESULT GetProcessProtocol ([out] VARIANT \* *pbstrPath*, [out] VARIANT \* *pvarResult*, [out] VARIANT \* *pvarErrorString*)**

**Beschreibung:** Diese Methode liefert das Prozessprotokoll.

**Parameter:**

*VARIANT*: hier wird der Pfad zurückgeliefert, in dem das Protokoll liegt

*VARIANT*: hier wird der Fehlercode zurückgeliefert, 0 -> kein Fehler aufgetreten

*VARIANT*: hier wird der Fehlertext zurückgeliefert

**Hinweis:** verfügbar bei Aufruf von

- [IWFCClient::GetProcessesFromModel\(\)](#)
- [IWFCClient::GetProcessesFromRole\(\)](#)
- [IWFCClient::GetProcessesFromUser\(\)](#)
- [IWFCClient::GetHistProcessesFromModel\(\)](#)

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

**Beschreibung:** Diese Methode liefert den Endzeitpunkt des Prozesses in Form eines Zeitstempels.

**Rückgabewerte:**

*long*: Zeitpunkt der Prozessbeendigung

**Hinweis:** verfügbar bei Aufruf von

- [IWFCClient::GetHistProcessesFromModel\(\)](#)

### **HRESULT DePersonalize([in] BSTR bstrUserId, [out]VARIANT\* pvarResult, [out]VARIANT \*pvarErrorString)**

**Beschreibung:** Diese Methode macht die Personalisierung des Arbeitsschrittes rückgängig.

**Parameter:**

*BSTR*: Id des Benutzers, für den Arbeitsschritt personalisiert ist

*VARIANT*: hier wird der Fehlercode zurückgeliefert, 0 -> kein Fehler aufgetreten

*VARIANT*: hier wird ggf. der Fehlertext zurückgeliefert

**Hinweis:** verfügbar bei Aufruf von

- [IWFCClient::GetProcessesFromUser\(\)](#)

### **HRESULT Personalize([in] BSTR bstrUserId, [in] BSTR bstrUserName, [out]VARIANT\* pvarResult, [out]VARIANT \*pvarErrorString)**

**Beschreibung:** Diese Methode personalisiert den Arbeitsschritt für den angegebenen Benutzer.

**Parameter:**

*BSTR*: Id des Benutzers, für den Arbeitsschritt personalisiert werden soll

*BSTR*: Name des Benutzers, für den Arbeitsschritt personalisiert werden soll

*VARIANT*: hier wird der Fehlercode zurückgeliefert, 0 -> kein Fehler aufgetreten

*VARIANT*: hier wird ggf. der Fehlertext zurückgeliefert

**Hinweis:** verfügbar bei Aufruf von

- [IWFCClient::GetProcessesFromUser\(\)](#)

### **HRESULT PreSuspendState ([out, retval] long\* pIVal)**

**Beschreibung:** Diese Methode liefert den Status des Prozesses, bevor er angehalten wurde.

**Parameter:**

*LONG:* [Status](#) vor dem Anhalten

**Hinweis:** verfügbar bei Aufruf von

[IWFCClient::GetProcessesFromUser\(\)](#)

### **HRESULT ResumeProcess ([out]VARIANT\* pvarResult, [out]VARIANT \*pvarErrorString)**

**Beschreibung:** Diese Methode setzt einen angehaltenen Prozess wieder fort.

**Rückgabewerte:**

VARIANT: 0 = Prozess erfolgreich fortgesetzt, ansonsten Fehlernummer

VARIANT: ggf. Fehlertext

**Hinweis:** verfügbar bei Aufruf von

[IWFCClient::GetProcessesFromUser\(\)](#)

### **HRESULT SuspendProcess ([out]VARIANT\* pvarResult, [out]VARIANT \*pvarErrorString)**

**Beschreibung:** Diese Methode hält den Prozess an. Es können keine Arbeitsschritte mehr bearbeitet werden, sie werden aus dem Eingangskorb entfernt.

**Rückgabewerte:**

VARIANT: 0 = Prozess erfolgreich angehalten, ansonsten Fehlernummer

VARIANT: ggf. Fehlertext

**Hinweis:** verfügbar bei Aufruf von

[IWFCClient::GetProcessesFromUser\(\)](#)

## IWFAProcesses Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Liste von Prozessen. Die Schnittstelle wird vom enaio® - Workflow Administrator genutzt.

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

## IWFAProcessLock Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt einen vom System gelockten Prozess. Die Schnittstelle wird vom enaio® - Workflow Administrator verwendet. Zugriff auf diese Schnittstelle über:

- [IWFCClient::WFAProcessLocks\(\)](#)

### Öffentliche Methoden

**HRESULT [LockTime](#)**([out, retval] long \*pVal)  
**HRESULT [ProcessID](#)**([out, retval] BSTR \*pVal)  
**HRESULT [ProcessName](#)**([out, retval] BSTR \*pVal)  
**HRESULT [ServerID](#)**([out, retval] long \*pVal)  
**HRESULT [ThreadID](#)**([out, retval] long \*pVal)

### Dokumentation der Elementfunktionen

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

**Beschreibung:** Diese Methode gibt an, wann der Prozess gelockt worden ist.

**Rückgabewerte:**

*LONG:* Zeitstempel

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

**Beschreibung:** Diese Methode liefert die ID des Prozesses.

**Rückgabewerte:**

*BSTR:* ID des Prozesses

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

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

**Rückgabewerte:**

*BSTR*: Name des Processes

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

**Beschreibung:** Diese Methode liefert die ID des Servers, durch den der Prozess gelockt wurde.

**Rückgabewerte:**

*LONG*: ServerID

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

**Beschreibung:** Diese Methode liefert die ID des Threads, durch den der Prozess gelockt wurde.

**Rückgabewerte:**

*LONG*: ThreadID

## IWFAProcessLocks Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Liste mit Informationen zu gelockten Workflow-Prozessen. Die Schnittstelle wird vom enaio® - Workflow Administrator genutzt. Zugriff auf diese Schnittstelle über:

- [IWFCClient::WFAProcessLocks\(\)](#)

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

## IWFA LockInfoEntry Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle liefert Informationen zu einer Workflow-Datenbanktabelle die vom System gelockt wird. Die Schnittstelle wird vom enaio® - Workflow Administrator verwendet. Zugriff auf diese Schnittstelle über:

- [IWFCClient::WFALockInfoEntries\(\)](#)

### Öffentliche Methoden

**HRESULT** [LockTime](#)([out, retval] long \*pVal)  
**HRESULT** [ServerID](#)([out, retval] long \*pVal)  
**HRESULT** [State](#)([out, retval] long \*pVal)  
**HRESULT** [TableID](#)([out, retval] long \*pVal)  
**HRESULT** [TableName](#)([out, retval] BSTR \*pVal)  
**HRESULT** [ThreadID](#)([out, retval] long \*pVal)

### Dokumentation der Elementfunktionen

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

**Beschreibung:** Diese Methode gibt an, wann die Datenbank-Tabelle gelockt worden ist.

**Rückgabewerte:**

*LONG:* Zeitstempel

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

**Beschreibung:** Diese Methode liefert die ID des Servers, durch den die Tabelle gelockt wurde.

**Rückgabewerte:**

*LONG:* ID des Servers



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

**Beschreibung:** Diese Methode gibt an, ob die Datenbank-Tabelle gelockt ist.

**Rückgabewerte:**

*LONG:* 1 = gelockt, ansonsten 0

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

**Beschreibung:** Diese Methode liefert die Id der Datenbank-Tabelle.

**Rückgabewerte:**

*LONG:* Id der Datenbank-Tabelle

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

**Beschreibung:** Diese Methode liefert den Namen der Datenbank-Tabelle.

**Rückgabewerte:**

*BSTR:* Name der Datenbank-Tabelle

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

**Beschreibung:** Diese Methode liefert die Id des Threads, durch den die Tabelle gelockt wurde.

**Rückgabewerte:**

*LONG:* ThreadID

## IWFALockInfoEntries Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Liste mit Informationen zu Workflow-Datenbanktabellen die vom System gelockt werden. Die Schnittstelle wird vom enaio® - Workflow Administrator genutzt. Zugriff auf diese Schnittstelle über:

- [IWFCClient::WFALockInfoEntries\(\)](#)

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

## IWFARole Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Rolle. Die Schnittstelle wird vom enaio® - Workflow Administrator genutzt. Zugriff auf diese Schnittstelle über:

- [IWFClient::WFARoles\(\)](#)

### Öffentliche Methoden

**HRESULT [Count](#) ([out, retval] long \*pVal)**  
**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 aus dem Workflowmodell.

**Rückgabewerte:**

*BSTR*: Id der Rolle

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

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

**Rückgabewerte:**

*BSTR*: Name der Rolle

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

**Beschreibung:** Diese Methode gibt an, in wieviel laufenden Prozessen die Rolle verwendet wird.

**Rückgabewerte:**

*long*: Anzahl, Zuordnung Rolle -> Prozesse

## IWFARoles Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Liste von Rollen. Die Schnittstelle wird vom enaio® - Workflow Administrator genutzt. Zugriff auf diese Schnittstelle über:

- [IWFCClient::WFARoles\(\)](#)

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

## IWFAUser Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt einen Benutzer. Die Schnittstelle wird vom enaio® - Workflow Administrator genutzt. Zugriff auf diese Schnittstelle über:

- [IWFCClient::WFAUsers\(\)](#)
- [IWFCClient::GetHistWorkItemRelUsersFromProcess\(\)](#)

### Öffentliche Methoden

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

**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 aus dem Workflowmodell.

**Rückgabewerte:**

*BSTR:* Id des Benutzers

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

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

**Rückgabewerte:**

*BSTR:* Name des Benutzers

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

**Beschreibung:** Diese Methode gibt an, wieviel laufenden Prozessen der Benutzer zugeordnet ist.

**Rückgabewerte:**

*long:* Anzahl, Zuordnung Benutzer -> Prozesse

**Hinweis:** bei Aufruf über [IWFCClient::GetHistWorkItemRelUsersFromProcess](#) () nicht gesetzt

## IWFAUsers Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Liste von Benutzern. Die Schnittstelle wird vom enaio® - Workflow Administrator genutzt. Zugriff auf diese Schnittstelle über:

- [IWFCClient::WFAUsers\(\)](#)
- [IWFCClient::GetHistWorkItemRelUsersFromProcess\(\)](#)

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

## IWFWorkerqueueEntries Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Liste mit Informationen zu Workerqueue-Elementen. Die Schnittstelle wird vom enaio® - Workflow Administrator genutzt. Zugriff auf diese Schnittstelle über:

- [IWFCClient::WFAWorkerqueueEntries\(\)](#)

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

## IWFAWorkerqueueEntry Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt ein Workerqueue-Element (Aktivität die von der Workerqueue bearbeitet wird). Die Schnittstelle wird vom enaio® - Workflow Administrator genutzt. Zugriff auf diese Schnittstelle über:

- [IWFCClient::WFAWorkerqueueEntries\(\)](#)

### Öffentliche Methoden

**HRESULT [ActivityName](#)([out, retval] BSTR \*pVal)**  
**HRESULT [ActivityState](#)([out, retval] long \*pVal)**  
**HRESULT [CreationTime](#)([out, retval] long \*pVal)**  
**HRESULT [LockState](#)([out, retval] long \*pVal)**  
**HRESULT [LockTime](#)([out, retval] long \*pVal)**  
**HRESULT [ProcessName](#)([out, retval] BSTR \*pVal)**  
**HRESULT [RActivityID](#)([out, retval] BSTR \*pVal)**  
**HRESULT [ServerID](#)([out, retval] long \*pVal)**  
**HRESULT [TargetState](#)([out, retval] long \*pVal)**

### Dokumentation der Elementfunktionen

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

**Beschreibung:** Diese Methode liefert die Aktivitätenname des Workerqueue-Elements.

**Rückgabewerte:**

*BSTR*: Name der Aktivität

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

**Beschreibung:** Diese Methode liefert die Aktivitätenstatus des Workerqueue-Elements.

**Rückgabewerte:**

*LONG*: [Aktivitätenstatus](#)



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

**Beschreibung:** Diese Methode liefert den Zeitpunkt der Erstellung des Workerqueue-Elements.

**Rückgabewerte:**

*LONG*: Erstellzeit

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

**Beschreibung:** Diese Methode gibt an, ob das Workerqueue-Element gelockt ist.

**Rückgabewerte:**

*LONG*: 1 = gelockt, ansonsten 0

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

**Beschreibung:** Diese Methode gibt an, wann das Workerqueue-Element gelockt worden ist.

**Rückgabewerte:**

*LONG*: Zeitstempel

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

**Beschreibung:** Diese Methode liefert den Prozessname zum Workerqueue-Element.

**Rückgabewerte:**

*BSTR*: Prozessname

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

**Beschreibung:** Diese Methode liefert die Instanzid des Workerqueue-Elements.

**Rückgabewerte:**

*BSTR*: Instanzid

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

**Beschreibung:** Diese Methode liefert die Id des Servers, der das Workerqueue-Element bearbeitet.

**Rückgabewerte:**

*LONG:* Id des Servers

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

**Beschreibung:** Diese Methode liefert den Status, den das Workerqueue-Element nach Bearbeitung durch die Workerqueue erreichen soll.

**Rückgabewerte:**

*LONG:* [Status](#) der Aktivität nach Bearbeitung durch Workerqueue

## 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. Um alle Methoden (z.B. [IsWFUser\(\)](#)) dieser Schnittstelle nutzen zu können, muß als erstes mit Hilfe der Methode [Connect](#) () eine Verbindung zum Workflowsystem hergestellt werden.

### Öffentliche Methoden

**HRESULT [ActiveOrganisation](#)** ([out, retval] [IOrganisation](#) \*\*pVal)  
**HRESULT [CurrentUserAbsent](#)** ([out, retval] VARIANT\_BOOL \*pVal)  
**HRESULT [GetActivityPerformers](#)** ([in]BSTR bstrActivityID,[out, retval][IWFActivityPerformers](#) \*\*ppActivityPerformers)  
**HRESULT [GetProcessProtocol](#)** ([in]BSTR bstrProcessId,[out]VARIANT \*pbstrPath,[out]VARIANT \*pvarResult,[out]VARIANT \*pvarErrorString)  
**HRESULT [GetVariablesFromActivity](#)** ([in]BSTR bstrActivityID,[out]VARIANT \*pvarResult,[out]VARIANT \*pvarErrorString,[out, retval][IWFVariables](#) \*\*ppVariables)  
**HRESULT [GetWFProcessModelByModelIdVersion](#)**([in] BSTR bstrModelId, [in] long lVersion, [out,retval] IWFProcessModel\*\* pProcessModel);  
**HRESULT [IsWFUser](#)** ([out, retval] VARIANT\_BOOL \*pVal)  
**HRESULT [RefreshOrganisation](#)** ()  
**HRESULT [RefreshWorkItemList](#)** ([in] VARIANT\_BOOL bRefresh)  
**HRESULT [RefreshWorkItemList](#)** ([out, retval] VARIANT\_BOOL \*pbRefresh)  
**HRESULT [RunningActivities](#)** ([out, retval] [IWFActivities](#) \*\*ppActivities)  
**HRESULT [SetUserAbsence](#)** ([in] VARIANT\_BOOL bAbsent,[out]VARIANT \*pvarErrorString,[out, retval]long \*lpResult)  
**HRESULT [SetVariablesFromActivity](#)** ([in]BSTR bstrActivityID,[out]VARIANT \*pvarResult,[out]VARIANT \*pvarErrorString,[in][IWFVariables](#) \*pVariables)  
**HRESULT [OrgUserID](#)**([out, retval] BSTR\* pVal)  
**HRESULT [UserName](#)** ([out, retval] BSTR \*pVal)  
**HRESULT [UserID](#)**([out, retval] BSTR\* pVal)  
**HRESULT [WFItems](#)** ([out, retval] [IWFItems](#) \*\*pVal)  
**HRESULT [AllWFItems](#)** ([out, retval] [IWFItems](#) \*\*pVal)  
**HRESULT [WFOrganisations](#)** ([out, retval][IWFOrganisations](#) \*\*ppOrganisations)  
**HRESULT [WFWorkItems](#)** ([out, retval] IWFWorkItems \*\*ppWorkItems)  
**HRESULT [UpdateWorkitemList](#)**([out, retval] VARIANT\_BOOL \*pVal)  
**HRESULT [RefreshWorkItemsAsync](#)**()  
**HRESULT [MaxAddInboxItemsNotifcationSize](#)**([out, retval] long \* lSize)  
**HRESULT [RefreshRunningActivitiesAsync](#)**()  
**HRESULT [GetProcessFile](#)**([in] BSTR ProcessID, [out] VARIANT\* pvarResult, [out] VARIANT\* pvarErrorString, [out,retval]IWFFile \*\*pWFFile)

#### Methoden für Verbindungsaufbau zum Workflowsystem

Diese Methoden werden für den Verbindungsaufbau zum Workflowsystem verwendet.

**HRESULT [Connect](#)** ([out, retval] VARIANT\_BOOL \*pbResult)  
**HRESULT [Connected](#)** ([out, retval] VARIANT\_BOOL \*pVal)  
**HRESULT [Disconnect](#)** ()  
**HRESULT [UseRapidTranslation](#)** ([in] VARIANT\_BOOL newVal)  
**HRESULT [UseRapidTranslation](#)** ([out, retval] VARIANT\_BOOL \*pVal)

#### Methoden für den enaio® - Workflow Administrator

Diese Methoden ermöglichen den Zugriff auf die aktuellen Prozesse.

**HRESULT [GetActivitiesFromProcess](#)** ([in]BSTR bstrProcessID,[out]VARIANT \*pvarResult,[out]VARIANT \*pvarErrorString,[out, retval][IWFAActivities](#) \*\*ppActivities)  
**HRESULT [GetProcessesFromModel](#)** ([in]BSTR bstrModelID,[out]VARIANT \*pvarResult,[out]VARIANT \*pvarErrorString,[out, retval][IWFAProcesses](#) \*\*ppProcesses)  
**HRESULT [GetProcessesFromRole](#)** ([in]BSTR bstrRoleID,[out]VARIANT \*pvarResult,[out]VARIANT \*pvarErrorString,[out, retval][IWFAProcesses](#) \*\*ppProcesses)  
**HRESULT [GetProcessesFromUser](#)** ([in]BSTR bstrUserID,[out]VARIANT \*pvarResult,[out]VARIANT \*pvarErrorString,[out, retval][IWFAProcesses](#) \*\*ppProcesses)  
**HRESULT [RollbackProcess](#)** ([in]BSTR bstrRActivityID, [in]long lExecPoint, [in]VARIANT\_BOOL bDoRollback, [out, retval] [IWFAActivities](#) \*\* ppActivities)  
**HRESULT [WFAModels](#)** ([out, retval][IWFAModels](#) \*\*ppModels)  
**HRESULT [WFARoles](#)** ([out, retval][IWFARoles](#) \*\*ppRoles)  
**HRESULT [WFAUsers](#)** ([out, retval][IWFAUsers](#) \*\*ppUsers)  
**HRESULT [WFAWorkerqueueEntries](#)**([out, retval] [IWFAWorkerqueueEntries](#)\*\* ppWQEntries)  
**HRESULT [WFAProcessLocks](#)**([out, retval] IWFAProcessLocks\* \*pVal)  
**HRESULT [WFALockInfoEntries](#)**([out, retval] IWFALockInfoEntries\* \*pVal)  
**HRESULT [AdminReleaseLock](#)**([in] long lLocktype, [in] BSTR bstrLockID)

#### Methoden für die Verwaltung von Adhoc-Lauflisten-Templates

Diese Methoden ermöglichen die Verwaltung von Adhoc-Lauflisten-Templates.

**HRESULT [GetAdhocTemplateList](#)**([in]BSTR bstrOrgID,[ out, retval][IRoutingLists](#) \*\*ppTemplates)  
**HRESULT [DeleteAdhocTemplate](#)** ([in]BSTR bstrOrgID,[in]BSTR bstrTemplateId)  
**HRESULT [PublishAdhocTemplate](#)** ([in]BSTR bstrOrgID,[in]BSTR bstrTemplateId)  
**HRESULT [UnpublishAdhocTemplate](#)** ([in]BSTR bstrOrgID,[in]BSTR bstrTemplateId)  
**HRESULT [SaveAdhocTemplate](#)** ([in]BSTR bstrOrgID,[in]BSTR bstrTemplateId, [in]BSTR bstrTemplateName, [in]VARIANT\_BOOL bTemplateIsPublic, [in] [IRoutingList](#)\* pRoutingList, [out, retval] BSTR \*pbstrOutTemplateId)  
**HRESULT [CheckAdhocTemplate](#)**([in] BSTR bstrTemplateId, [in] BSTR bstrRActivityId, [out, retval] VARIANT\_BOOL\* pbIsValid)  
**HRESULT [GetAdhocTemplate](#)**([in] BSTR bstrOrgId, [in] BSTR bstrTemplateId, [out,retval] [IRoutingList](#)\*\* ppRoutingList);

### Methoden für Historienverwaltung

Diese Methoden werden vom enaio® - Workflow Administrator für die Historienverwaltung verwendet.

**HRESULT [GetHistActivitiesFromProcess](#)** ([in] BSTR bstrProcessID,[out, retval][IWFAHistActivities](#) \*\*ppHistActivities)  
**HRESULT [GetHistEntriesFromProcess](#)** ([in] BSTR bstrProcessID,[out, retval][IWFAHistEntries](#) \*\*ppHistEntries)  
**HRESULT [GetHistEntriesFromRActivity](#)** ([in] BSTR bstrRActivityID,[out, retval][IWFAHistEntries](#) \*\*ppHistEntries)  
**HRESULT [GetHistEntriesFromTimer](#)** ([in] BSTR bstrTimerID,[out, retval][IWFAHistEntries](#) \*\*ppHistEntries)  
**HRESULT [GetHistProcessesFromModel](#)** ([in] BSTR bstrModelID,[out, retval][IWFAProcesses](#) \*\*ppProcesses)  
**HRESULT [GetHistTimersFromProcess](#)** ([in] BSTR bstrProcessID,[out, retval][IWFAHistTimers](#) \*\*ppHistTimers)  
**HRESULT [GetHistVariablesByHistEntry](#)** ([in]BSTR bstrHistID,[out]VARIANT \*pvarResult,[out]VARIANT \*pvarErrorString,[out, retval][IWFFVariables](#) \*\*ppVariables)  
**HRESULT [GetHistWorkItemRelActivitiesFromProcess](#)** ([in] BSTR bstrProcessID,[out, retval][IWFAHistActivities](#) \*\*ppHistActivities)  
**HRESULT [GetHistWorkItemRelEntriesByActivity](#)** ([in] BSTR bstrRActivityID,[out, retval][IWFAHistWorkItemRelEntries](#) \*\*ppHistWorkItemRelEntries)  
**HRESULT [GetHistWorkItemRelEntriesByUser](#)** ([in] BSTR bstrProcessID,[in] BSTR bstrUserID,[out, retval][IWFAHistWorkItemRelEntries](#) \*\*ppHistWorkItemRelEntries)  
**HRESULT [GetHistWorkItemRelUsersFromProcess](#)** ([in] BSTR bstrProcessID,[out, retval][IWFAUsers](#) \*\*ppUsers)  
**HRESULT [HistWorkflows](#)** ([out, retval][IWFAHistWorkflows](#) \*\*ppHistWorkflows)

### Dokumentation der Elementfunktionen

**HRESULT Connect** ([out, retval] VARIANT\_BOOL \* *pbResult*)

**Beschreibung:** Diese Methode stellt eine Verbindung zum Workflowsystem her. Die Methode setzt ein Login durch die AXBASICS.DLL (enaio®-Anmeldung) voraus. Wurde kein solches Login durchgeführt, erfolgt die Anmeldung ans enaio® - System durch diese Methode.

**Rückgabewerte:**

*VARIANT\_BOOL*: TRUE, wenn die Verbindung erfolgreich hergestellt werden konnte.

### **HRESULT Disconnect ()**

**Beschreibung:** Diese Methode unterbricht die Verbindung zum Workflowsystem.

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

**Beschreibung:** Diese Methode gibt an, ob die Anwendung RapidTranslation nutzt.

**Rückgabewerte:**

*VARIANT\_BOOL*: TRUE = RapidTranslation wird genutzt

### **HRESULT UseRapidTranslation ([in] VARIANT\_BOOL newVal)**

**Beschreibung:** Diese Methode setzt die Eigenschaft RapidTranslation. Diese Eigenschaft sollte dann gesetzt werden, wenn noch kein Login über die AXBASICS.DLL erfolgt ist. Projekte, die kein ‚RapidTranslation‘ nutzen (z.B. VB-Programme), müssen die Eigenschaft generell auf FALSE setzen.

**Parameter:**

*VARIANT\_BOOL*: TRUE = RapidTranslation wird genutzt

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

**Beschreibung:** Diese Methode gibt an, ob eine Verbindung zum Workflowsystem besteht.

**Rückgabewerte:**

*VARIANT\_BOOL*: TRUE = Verbindung besteht

### **HRESULT WFWorkItems ([out, retval] IWFWorkItems \*\* ppWorkItems)**

**Beschreibung:** Diese Methode liefert für den aktuell angemeldeten Benutzer eine Liste aller eingegangenen Vorgänge/Vorgangsschritte. Hiermit kann ein Eingangskorb realisiert werden.

**Rückgabewerte:**

*IWFWorkItems*: Liste aller Vorgänge

### **HRESULT WFItems ([out, retval] IWFItems \*\* pVal)**

**Beschreibung:** Diese Methode liefert für den aktuell angemeldeten Benutzer alle startbaren Vorgänge bzw. Workflowprozesse.

**Rückgabewerte:**

*IWFItems*: Liste aller startbaren Vorgänge

### **HRESULT AllWFItems ([out, retval] IWFItems \*\* pVal)**

**Beschreibung:** Diese Methode liefert alle startbaren Vorgänge bzw. Workflowprozesse. Es werden auch die Workflowprozesse ermittelt, die von diesem Benutzer nicht gestartet werden können.

**Rückgabewerte:**

*IWFItems*: Liste aller startbaren Vorgänge

### **HRESULT RefreshWorkItemList ([out, retval] VARIANT\_BOOL \* *pbRefresh*)**

**Beschreibung:** Diese Methode gibt an, ob die Liste der eingegangenen Vorgänge erneuert werden soll.

**Rückgabewerte:**

*VARIANT\_BOOL*: TRUE = Liste erneuern

Siehe auch: [IWFWorkItems](#)

### **HRESULT RefreshWorkItemList ([in] VARIANT\_BOOL *bRefresh*)**

**Beschreibung:** Diese Methode bestimmt, ob der Eingangskorb erneuert werden soll. Wird diese Eigenschaft auf TRUE gesetzt, wird die Liste aller eingegangenen Vorgänge/Vorgangsschritte neu beim Archivserver angefordert. Dann sollte man den Eingangskorb neu abfragen. Ist diese Eigenschaft auf FALSE gesetzt, wird bei jeder Anforderung des Eingangskorbs, der bereits geladene zurückgegeben.

Siehe auch: [IWFWorkItems](#)

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

**Beschreibung:** Diese Eigenschaft gibt an, ob der Server bei Änderungen des Eingangskorbes Benachrichtigungen verschickt. Die Einstellung dieser Eigenschaft erfolgt in der Serverregistrierung.

**Rückgabewerte:**

*VARIANT\_BOOL*: TRUE = Server verschickt Benachrichtigungen

### **HRESULT RefreshWorkItemsAsync()**

**Beschreibung:** Diese Methode informiert den Server darüber, dass der Client eine asynchrone Aktualisierung des Eingangskorbes anfordert.

### **HRESULT MaxAddInboxItemsNotifcationSize([out, retval] long \* *ISize*)**

**Beschreibung:** Diese Eigenschaft beschreibt die maximale Anzahl an Arbeitsschritten im Eingangskorb für eine asynchrone Benachrichtigung des Clients. Der Wert wird über die Serverregistrierung gesetzt.

### **HRESULT RefreshRunningActivitiesAsync()**

**Beschreibung:** Diese Methode informiert den Server darüber, dass der Client eine asynchrone Aktualisierung der laufenden Vorgänge anfordert.

### **HRESULT GetProcessFile([in] BSTR *ProcessID*, [out] VARIANT\* *pvarResult*, [out] VARIANT\* *pvarErrorString*, [out,retval]IWFFile \*\**pWFFFile*)**

**Beschreibung:** Diese Methode liefert für einen Prozess die aktuelle Akte.

**Parameter:**

*BSTR*: Id des Prozesses

*VARIANT*: Ergebnis des Jobaufrufes

*VARIANT*: Fehlertext

**Rückgabewerte:**

*IWFFile*: Workflowakte

**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

**HRESULT WFAModels ([out, retval] [IWFAModels](#) \*\* *ppModels*)**

**Beschreibung:** Diese Methode liefert eine Liste aller aktiven Workflow-Modelle.

**Rückgabewerte:**

*IWFAModels*: Liste aller aktiven Workflow-Modelle



### **HRESULT WFARoles ([out, retval] [IWFARoles](#) \*\* ppRoles)**

**Beschreibung:** Diese Methode liefert eine Liste aller Rollen, die in aktuellen Vorgängen verwendet werden.

**Rückgabewerte:**

*IWFARoles:* Liste aller aktiven Rollen

### **HRESULT WFAUsers ([out, retval] [IWFAUsers](#) \*\* ppUsers)**

**Beschreibung:** Diese Methode liefert eine Liste aller Benutzer, die aktuellen Vorgängen zugeordnet sind.

**Rückgabewerte:**

*IWFAUsers:* Liste aller Benutzer

### **HRESULT GetProcessesFromModel ([in] BSTR bstrModelID, [out] VARIANT \* pvarResult, [out] VARIANT \* pvarErrorString, [out, retval] [IWFAProcesses](#) \*\* ppProcesses)**

**Beschreibung:** Diese Methode liefert eine Liste aller Prozesse zu einem bestimmten Workflow-Modell.

**Parameter:**

*BSTR:* ID des Models von dem die Liste geholt werden soll

*VARIANT:* der Fehlercode wird hier gespeichert. 0, wenn kein Fehler aufgetreten ist

*VARIANT:* der Fehlertext

**Rückgabewerte:**

*IWFAProcesses:* Liste aller Prozesse

### **HRESULT GetProcessesFromUser ([in] BSTR bstrUserID, [out] VARIANT \* pvarResult, [out] VARIANT \* pvarErrorString, [out, retval] [IWFAProcesses](#) \*\* ppProcesses)**

**Beschreibung:** Diese Methode liefert eine Liste aller Aktivitäten im Eingangskorb eines bestimmten Benutzers. Das Rückgabe-Objekt enthält auch Prozessinformationen.

**Parameter:**

*BSTR:* ID des Users von dem die Liste geholt werden soll

*VARIANT:* der Fehlercode wird hier gespeichert. 0, wenn kein Fehler aufgetreten ist

*VARIANT:* der Fehlertext

**Rückgabewerte:**

*IWFAProcesses:* Liste aller Aktivitäten im Eingangskorb mit Prozessinformationen

### **HRESULT GetProcessesFromRole ([in] BSTR bstrRoleID, [out] VARIANT \* pvarResult, [out] VARIANT \* pvarErrorString, [out, retval] [IWFAProcesses](#) \*\* ppProcesses)**

**Beschreibung:** Diese Methode liefert eine Liste aller Aktivitäten im Eingangskorb einer bestimmten Rolle. Das Rückgabe-Objekt enthält auch Prozessinformationen

**Parameter:**

*BSTR*: ID der Rolle von der die Liste geholt werden soll  
*VARIANT*: der Fehlercode wird hier gespeichert 0, wenn kein Fehler aufgetreten ist  
*VARIANT*: der Fehlertext

**Rückgabewerte:**

*IWFAProcesses*: Liste aller Aktivitäten im Eingangskorb mit Prozessinformationen

**HRESULT GetActivitiesFromProcess ([in] BSTR *bstrProcessID*, [out] VARIANT \*  
*pvarResult*, [out] VARIANT \* *pvarErrorString*, [out, retval] [IWFAActivities](#) \*\*  
*ppActivities*)**

**Beschreibung:** Diese Methode liefert eine Liste der Vorgangsschritte eines Prozesses von der Start-Aktivität bis zum aktuellen Vorgangsschritt im Eingangskorb.

**Parameter:**

*BSTR*: ID des Processes von dem die Liste geholt werden soll  
*VARIANT*: der Fehlercode wird hier gespeichert. 0, wenn kein Fehler aufgetreten ist  
*VARIANT*: der Fehlertext

**Rückgabewerte:**

*IWFAActivities*: Liste von Vorgangsschritten

**HRESULT GetVariablesFromActivity ([in] BSTR *bstrActivityID*, [out] VARIANT \* *pvarResult*, [out] VARIANT \* *pvarErrorString*, [out, retval] [IWFVariables](#) \*\* *ppVariables*)**

**Beschreibung:** Diese Methode liefert eine Liste aller Variablen einer beliebigen Aktivität.

**Parameter:**

*BSTR*: Instanzid der Aktivität, von der die Liste erstellt werden soll

*VARIANT*: der Fehlercode wird hier gespeichert. Null, wenn kein Fehler aufgetreten ist

*VARIANT*: der Fehlertext

**Rückgabewerte:**

*IWFVariables*: Liste aller Variablen

**HRESULT SetVariablesFromActivity ([in] BSTR *bstrActivityID*, [out] VARIANT \* *pvarResult*, [out] VARIANT \* *pvarErrorString*, [in] [IWFVariables](#) \* *pVariables*)**

**Beschreibung:** Diese Methode setzt die Variablen zu einer Aktivität.

**Parameter:**

*BSTR*: Instanzid der Aktivität, von der die Liste gesetzt werden soll

*VARIANT*: der Fehlercode wird hier gespeichert. Null, wenn kein Fehler aufgetreten ist

*VARIANT*: der Fehlertext

**Rückgabewerte:**

*IWFVariables*: Liste aller Variablen

**HRESULT GetProcessProtocol ([in] BSTR *bstrProcessId*, [out] VARIANT \* *pbstrPath*, [out] VARIANT \* *pvarResult*, [out] VARIANT \* *pvarErrorString*)**

**Beschreibung:** Diese Methode liefert das Prozessprotokoll.

**Parameter:**

*BSTR*: Prozessid

*VARIANT*: hier wird der Pfad zurückgeliefert, in dem das Protokoll liegt

*VARIANT*: hier wird der Fehlercode zurückgeliefert, 0 -> kein Fehler aufgetreten

*VARIANT*: hier wird der Pfad Fehlertext zurückgeliefert

**HRESULT GetWFProcessModelByModelIdVersion([in] BSTR *bstrModelId*, [in] long *lVersion*, [out,retval] [IWFProcessModel](#)\*\* *pProcessModel*);**

**Beschreibung:** Diese Methode liefert ein Workflowmodell anhand der ModellId und -Version.

**Parameter:**

*BSTR*: ModellId

*Long*: ModelVersion

**Rückgabewerte:**

*IWFProcessModel*: Modell

### **HRESULT WFOrganisations ([out, retval] [IWFOrganisations](#) \*\* ppOrganisations)**

**Beschreibung:** Diese Methode liefert eine Liste aller Organisationen, die im enaio® - Workflow Editor definiert wurden.

**Rückgabewerte:**

*IWFOrganisations:* Liste aller Organisationen

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

**Beschreibung:** Diese Methode gibt an, ob der aktuell angemeldete Benutzer in der Organisationsstruktur bekannt ist.

**Rückgabewerte:**

*VARIANT\_BOOL:* TRUE = Benutzer bekannt

### **HRESULT RunningActivities ([out, retval] [IWFActivities](#) \*\* ppActivities)**

**Beschreibung:** Diese Methode liefert eine Liste aller laufenden Aktivitäten, bei denen der angemeldete Benutzer als Prozessverantwortlicher eingetragen ist. Die Liste wird im Workflow für die Registerkarte "Laufende Vorgänge" verwendet.

**Rückgabewerte:**

*IWFActivities:* Liste aller laufenden Aktivitäten

### **HRESULT ActiveOrganisation ([out, retval] [IOrganisation](#) \*\* pVal)**

**Beschreibung:** Diese Methode liefert die aktive Organisationsstruktur.

**Rückgabewerte:**

*IOrganisation:* aktive Organisationsstruktur

### **HRESULT RefreshOrganisation ()**

**Beschreibung:** Diese Methode aktualisiert die aktive Organisationsstruktur.

### **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

**HRESULT HistWorkflows ([out, retval] [IWFAHistWorkflows](#) \*\*  
*ppHistWorkflows*)**

**Beschreibung:** Diese Methode liefert eine Liste aller historischen Workflows.

**Rückgabewerte:**

*IWFAHistWorkflows*: Liste aller historischen Workflows

**HRESULT GetHistProcessesFromModel ([in] BSTR *bstrModelID*, [out, retval]  
[IWFAProcesses](#) \*\* *ppProcesses*)**

**Beschreibung:** Diese Methode liefert eine Liste aller historischen Prozesse zu einem historischen Workflowmodell.

**Parameter:**

*BSTR*: historische Modell-Id

**Rückgabewerte:**

*IWFAProcesses*: Liste aller historischen Prozesse

**Hinweis:** historische Modell-Id siehe [HistWorkflows \(\)](#) -> [IWFAHistWorkflow::HistWFID \(\)](#)

**HRESULT GetHistActivitiesFromProcess ([in] BSTR *bstrProcessID*, [out, retval]  
[IWFAHistActivities](#) \*\* *ppHistActivities*)**

**Beschreibung:** Diese Methode liefert eine Liste aller Vorgangsschritte eines historischen Prozesses.

**Parameter:**

*BSTR*: historische Prozess-Id

**Rückgabewerte:**

*IWFAHistActivities*: Liste der Vorgangsschritte

**Hinweis:** historische Prozess-Id siehe [GetHistProcessesFromModel \(\)](#) -> [IWFAProcess::ID \(\)](#)

**HRESULT GetHistEntriesFromProcess ([in] BSTR *bstrProcessID*, [out, retval]  
[IWFAHistEntries](#) \*\* *ppHistEntries*)**

**Beschreibung:** Diese Methode liefert eine Liste aller historischen Einträge zu einem Prozess.

**Parameter:**

*BSTR*: historische Prozess-Id

**Rückgabewerte:**

*IWFAHistEntries*: Liste aller historischen Einträge

**Hinweis:** historische Prozess-Id siehe [GetHistProcessesFromModel \(\)](#) -> [IWFAProcess::ID \(\)](#)

**HRESULT GetHistEntriesFromRActivity ([in] BSTR *bstrRActivityID*, [out, retval]  
[IWFAHistEntries](#) \*\* *ppHistEntries*)**

**Beschreibung:** Diese Methode liefert eine Liste aller historischen Einträge zu einem bestimmten Vorgangs-schritt.

**Parameter:**

*BSTR*: Instanz-Id

**Rückgabewerte:**

*IWFAHistEntries*: Liste aller historischen Einträge

**Hinweis:** hist. Instanz-Id siehe [GetHistActivitiesFromProcess \(\)](#) -> [IWFAHistActivity::RActivityID \(\)](#)

**HRESULT GetHistTimersFromProcess ([in] BSTR *bstrProcessID*, [out, retval]  
[IWFAHistTimers](#) \*\* *ppHistTimers*)**

**Beschreibung:** Diese Methode liefert eine Liste aller historischen Mahn-/Sperrfristen eines Prozesses.

**Parameter:**

*BSTR*: hist. Prozess-Id

**Rückgabewerte:**

*IWFAHistTimers*: Liste aller historischen Mahn-/Sperrfristen

**Hinweis:** historische Prozess-Id siehe [GetHistProcessesFromModel \(\)](#) -> [IWFAProcess::ID \(\)](#)

**HRESULT GetHistEntriesFromTimer ([in] BSTR *bstrTimerID*, [out, retval] [IWFAHistEntries](#) \*\* *ppHistEntries*)**

**Beschreibung:** Diese Methode liefert eine Liste aller historischen Einträge zu einer Mahn-/Sperrfrist.

**Parameter:**

*BSTR*: Timer-Id

**Rückgabewerte:**

*IWFAHistEntries*: Liste aller historischen Einträge

**Hinweis:** historische Prozess-Id siehe [GetHistTimersFromProcess \(\)](#) -> [IWFAHistTimer::TimerID \(\)](#)

**HRESULT GetHistWorkItemRelUsersFromProcess ([in] BSTR *bstrProcessID*, [out, retval] [IWFAUsers](#) \*\* *ppUsers*)**

**Beschreibung:** Diese Methode liefert eine Liste aller User, die einen bestimmten historischen Prozess im Eingangskorb hatten.

**Parameter:**

*BSTR*: hist. Prozess-Id

**Rückgabewerte:**

*IWFAUsers*: Liste aller User

**Hinweis:** historische Prozess-Id siehe [GetHistProcessesFromModel \(\)](#) -> [IWFAProcess::ID \(\)](#)

**HRESULT GetHistWorkItemRelActivitiesFromProcess ([in] BSTR *bstrProcessID*, [out, retval] [IWFAHistActivities](#) \*\* *ppHistActivities*)**

**Beschreibung:** Diese Methode liefert eine Liste aller Aktivitäten eines historischen Prozesses, die als Vorgangsschritte im Eingangskorb zu sehen waren.

**Parameter:**

*BSTR*: hist. Prozess-Id

**Rückgabewerte:**

*IWFAHistActivities*: Liste aller Aktivitäten

**Hinweis:** historische Prozess-Id siehe [GetHistProcessesFromModel \(\)](#) -> [IWFAProcess::ID \(\)](#)

**HRESULT GetHistWorkItemRelEntriesByUser ([in] BSTR *bstrProcessID*, [in] BSTR *bstrUserID*, [out, retval] [IWFAHistWorkItemRelEntries](#) \*\* *ppHistWorkItemRelEntries*)**

**Beschreibung:** Diese Methode liefert eine Liste aller Eingangskorbeinträge eines bestimmten Benutzers zu einem historischen Prozess.

**Parameter:**

*BSTR*: hist. Prozess-Id

*BSTR*: Id des Benutzers

**Rückgabewerte:**

*ppHistWorkItemRelEntries*

**Hinweis:** historische Prozess-Id siehe [GetHistProcessesFromModel \(\)](#) -> [IWFAProcess::ID \(\)](#)

**HRESULT GetHistWorkItemRelEntriesByActivity ([in] BSTR *bstrRActivityID*, [out, retval] [IWFAHistWorkItemRelEntries](#) \*\* *ppHistWorkItemRelEntries*)**

**Beschreibung:** Diese Methode liefert eine Liste aller Eingangskorbeinträge einer historischen Aktivität.

**Parameter:**

*BSTR*: hist. Instanz-Id

**Rückgabewerte:**

*IWFAHistWorkItemRelEntries*: Liste aller Eingangskorbeinträge

**Hinweis:** hist. Instanz-Id siehe [GetHistActivitiesFromProcess \(\)](#) -> [IWFAHistActivity::RActivityID \(\)](#)

**HRESULT GetActivityPerformers ([in] BSTR *bstrRActivityID*, [out, retval] [IWFAActivityPerformers](#) \*\* *ppActivityPerformers*)**

**Beschreibung:** Diese Methode liefert eine Liste aller Teilnehmer zu einer bestimmten Aktivität.

**Parameter:**

*BSTR*: InstanzId der Aktivität

**Rückgabewerte:**

*IWFAActivityPerformers*: Liste aller Teilnehmer



**HRESULT GetHistVariablesByHistEntry ([in] BSTR *bstrHistID*, [out] VARIANT \* *pvarResult*, [out] VARIANT \* *pvarErrorString*, [out, retval] [IWFFVariables](#) \*\* *ppVariables*)**

**Beschreibung:** Diese Methode liefert eine Liste aller Workflowvariablen zu einem beliebigen historischen Eintrag.

**Parameter:**

*BSTR*: hist. Eintrags-Id

*VARIANT*: hier wird der Fehlercode zurückgeliefert, 0 -> keine Fehler aufgetreten

*VARIANT*: hier wird ggf. der Fehlertext zurückgeliefert

**Hinweis:** historische Eintrags-Id siehe [IWFAHistEntry::HistID \(\)](#)

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

**Beschreibung:** Diese Methode liefert den Namen des aktuell angemeldeten Benutzers.

**Rückgabewerte:**

*BSTR*: Benutzername

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

**Beschreibung:** Diese Methode liefert die ID des aktuell angemeldeten Benutzers.

**Rückgabewerte:**

*BSTR*: BenutzerID

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

**Beschreibung:** Diese Methode liefert die Organisationsobjekt ID des aktuell angemeldeten Benutzers.

**Rückgabewerte:**

*BSTR*: OrgObject BenutzerID

**HRESULT RollbackProcess([in]BSTR *bstrRActivityID*, [in]long *lExecPoint*, [in] VARIANT\_BOOL *bDoRollback*, [out, retval] [IWFAActivities](#)\*\* *ppActivities*)**

**Beschreibung:** Diese Methode setzt einen Prozess auf die angegebene Aktivität zurück und liefert alle Aktivitäten des Prozesses, die dadurch gelöscht werden.

**Parameter:**

*BSTR*: InstanzId der Aktivität, auf die der Prozess zurückgesetzt werden soll

*LONG*: Startpunkt an der Aktivität

100 = Aktivität ist erstellt

110 = nach Ausführung des StartActivity-Events

190 = vor Ausführung des EndActivity-Events

200 = Aktivität ist beendet

*VARIANT\_BOOL*: FALSE = Zurücksetzen wird nicht durchgeführt; TRUE = Prozess wird zurückgesetzt

**Rückgabewerte:**

*IWFAActivities*: Liste aller Aktivitäten die nach dem Zurücksetzen gelöscht wurden (DoRollback = TRUE) bzw. würden (DoRollback = FALSE)

**HRESULT WFAWorkerqueueEntries([out, retval] [IWFAWorkerqueueEntries](#)\*\*  
ppWQEntries)**

**Beschreibung:** Diese Methode liefert Informationen zu allen aktuellen Workerqueue-Elementen.

**Rückgabewerte:**

*IWFAWorkerqueueEntries:* Liste mit Informationen zu Workerqueue-Elementen

**Siehe auch:** [AdminReleaseLock](#)

**HRESULT WFAProcessLocks([out, retval] [IWFAProcessLocks](#)\* \*pVal)**

**Beschreibung:** Diese Methode liefert Informationen zu allen aktuell gelockten Workflowprozessen.

**Rückgabewerte:**

*IWFAWorkerqueueEntries:* Liste mit Informationen zu gelockten Workflowprozessen

**Siehe auch:** [AdminReleaseLock](#)

**HRESULT WFALockInfoEntries([out, retval] [IWFALockInfoEntries](#)\* \*pVal)**

**Beschreibung:** Diese Methode liefert Informationen zu allen Workflow-Datenbanktabellen die vom System gelockt werden.

**Rückgabewerte:**

*IWFAWorkerqueueEntries:* Liste mit Informationen zu gelockten Datenbanktabellen

**Siehe auch:** [AdminReleaseLock](#)

**HRESULT AdminReleaseLock([in] long lLocktype, [in] BSTR bstrLockID)**

**Beschreibung:** Diese Methode gibt ein Datenbanklock des angegebenen Typs wieder frei.

**Parameter:**

*LONG:* Locktyp

- 1 - gelockte Prozesse werden freigegeben ( bstrLockID muss dann entsprechende ProzessID enthalten
- 2 - gelocktes Workerqueue-Element wird freigegeben ( bstrLockID muss dann entsprechende WorkerqueueItemId enthalten
- 3 - gelockte Datenbanktabelle wird freigegeben ( bstrLockID muss dann entsprechende TabellenId enthalten

*BSTR:* Id des gelockten Elements (siehe Locktyp)

**HRESULT GetAdhocTemplateList([in]BSTR bstrOrgID,[ out, retval][IRoutingLists](#)  
\*\*ppTemplates)**

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

**Rückgabewerte:**

[IRoutingLists](#): Liste mit Vorlagen

**HRESULT DeleteAdhocTemplate ([in]BSTR bstrOrgID,[in]BSTR bstrTemplateId)**

**Beschreibung:** Diese Methode löscht eine Adhoc-Routinglisten-Vorlage.

**Parameter:**

*BSTR:* bstrOrgId – Id der Organisation

*BSTR*: bstrTemplateId – Id der zu löschenden Vorlage

### **HRESULT PublishAdhocTemplate ([in]BSTR bstrOrgID,[in]BSTR bstrTemplateId)**

**Beschreibung:** Diese Methode veröffentlicht eine Adhoc-Routinglisten-Vorlage. Ggf. im Speicher existierende Objekte dieses Templates werden nicht aktualisiert.

**Parameter:**

*BSTR*: bstrOrgId – Id der Organisation

*BSTR*: bstrTemplateId – Id der zu veröffentlichenden Vorlage

### **HRESULT UnpublishAdhocTemplate ([in]BSTR bstrOrgID,[in]BSTR bstrTemplateId)**

**Beschreibung:** Diese Methode privatisiert eine bisher öffentliche Adhoc-Routinglisten-Vorlage für den momentan angemeldeten Benutzer. Ggf. im Speicher existierende Objekte dieses Templates werden nicht aktualisiert.

**Parameter:**

*BSTR*: bstrOrgId – Id der Organisation

*BSTR*: bstrTemplateId – Id der zu privatisierenden Vorlage

### **HRESULT SaveAdhocTemplate ([in]BSTR bstrOrgID,[in]BSTR bstrTemplateId, [in]BSTR bstrTemplateName, [in]VARIANT\_BOOL bTemplateIsPublic, [in]IRoutingList\* pRoutingList, [out, retval] BSTR \*pbstrOutTemplateId)**

**Beschreibung:** Diese Methode speichert eine Adhoc-Routinglisten-Vorlage. Ggf. im Speicher existierende Templateobjekte werden nicht aktualisiert.

**Parameter:**

*BSTR*: bstrOrgId – Id der Organisation

*BSTR*: bstrTemplateId – Id der zu speichernden Vorlage; ist null wenn eine neue Vorlage erzeugt werden soll, ansonsten entspricht sie der Id der zu überschreibenden Vorlage

*BSTR*: bstrTemplateName – neuer Name der Vorlage

VARIANT\_BOOL: bTemplateIsPublic – gibt ab, ob die Vorlage öffentlich sein soll

IRoutingList\*: die zu speichernde Vorlage

**Rückgabewerte:**

*BSTR*: Id der gespeicherten Vorlage

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

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

**Parameter:**

*BSTR*: bstrTemplateId – Id der Vorlage

*BSTR*: bstrRAActivityId – Id des Arbeitsschrittes (Prozessaktivität)

**Rückgabewerte:**

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

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

**Beschreibung:** Diese Methode liefert eine Lauflistenvorlage anhand seiner Id.

**Parameter:**

*BSTR*: bstrOrgId – Id der Organisation

*BSTR*: bstrTemplateId – Id der Vorlage

**Rückgabewerte:**

IRoutingList\*: die Vorlage

## IWFEvent Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt einen Workflow-Event.

### Öffentliche Methoden

**HRESULT [ActivityID](#)** ([out, retval] BSTR \*pVal)  
**HRESULT [ID](#)** ([out, retval] BSTR \*pVal)  
**HRESULT [Parameter](#)** ([out, retval] BSTR \*pVal)  
**HRESULT [Script](#)** ([out, retval] [IWFScript](#) \*\*ppScript)  
**HRESULT [Type](#)** ([out, retval] \_WFEVENTTYPE \*pVal)  
**HRESULT [WorkflowID](#)** ([out, retval] BSTR \*pVal)

### Dokumentation der Elementfunktionen

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

**Beschreibung:** Diese Methode liefert den Eventtyp.

**Rückgabewerte:**

*long:* Eventtyp

**Hinweis:** mögliche Werte

- 0 = Unbekannt (UNKNOWN)
- 1 = Vor dem Weiterleiten (BEFOREFORWARD)
- 2 = Nach dem Weiterleiten (AFTERFORWARD)
- 3 = BEFOREFORWARDTO
- 4 = Button wurde gedrückt (BUTTONCLICK)
- 5 = Vor dem Öffnen der Datenmaske (BEFOREOPEN)
- 6 = nach digitaler Signatur eines Vorgangsschrittes (AFTERSIGNATURE)
- 7 = Vor dem Abbrechen eines Vorgangsschrittes (BEFORECANCEL)
- 8 = SIMULATEMASKEDIT,
- 10000 = Nach Erstellung der Aktivität am Server (STARTACTIVITY)
- 10001 = Vor dem Beenden der Aktivität am Server (ENDACTIVITY)
- 10002 = BEFORESTARTSUBPROC, ' Vor dem Starten eines Subprozesses am Server
- 10003 = Nachdem ein Benutzer einen nicht personalisierten Vorgangsschritt gestartet hat (PERSONALIZEWORKITEM)
- 10004 = GETWORKITEMPARAMS

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

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

**Rückgabewerte:**

*BSTR:* EventId

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

**Beschreibung:** Diese Methode liefert die Id des Buttons für das Event 'BUTTONCLICK'.

**Rückgabewerte:**

*BSTR:* Script

### **HRESULT Script ([out, retval] IWFScript \*\* ppScript)**

**Beschreibung:** Diese Methode liefert das zum Event hinterlegte Skript.

**Rückgabewerte:**

*IWFScript:* Skript

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

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

**Rückgabewerte:**

*BSTR:* Workflowid

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

**Beschreibung:** Diese Methode liefert die ID der Aktivität, der dieser Event zugeordnet ist.

**Rückgabewerte:**

*BSTR:* ID der Aktivität

## IWFEvents Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Liste aller definierten Events für eine bestimmte Aktivität.

Elemente der Liste: [IWFEvent](#)

### Öffentliche Methoden

**HRESULT [GlobalClientScript](#) ([out, retval] [IWFScript](#) \*\*pVal)**  
**HRESULT [GlobalServerScript](#) ([out, retval] [IWFScript](#) \*\*pVal)**

### Dokumentation der Elementfunktionen

**HRESULT GlobalClientScript ([out, retval] [IWFScript](#) \*\* pVal)**

**Beschreibung:** Diese Methode liefert das globale Client-Skript.

**Rückgabewerte:**

*IWFScript*: globales Client-Skript

**HRESULT GlobalServerScript ([out, retval] [IWFScript](#) \*\* pVal)**

**Beschreibung:** Diese Methode liefert das globale Server-Skript.

**Rückgabewerte:**

*IWFScript*: globales Server-Skript

## IWFFieldValues Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Auswahlliste an Textfeldern in Masken. Listen an Textfeldern werden im OS:4.x-Workflow\_Editor erstellt und gefüllt. Über diese Schnittstelle ist es möglich die Listeninhalte dynamisch in Events zu verändern. 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 [GetStringValue](#) ([in] BSTR strSection,[in] BSTR strEntry,[in] BSTR strDefault,[out, retval] BSTR \*strValue)**

**HRESULT [WriteStringValue](#) ([in] BSTR strSection,[in] BSTR strEntry,[in] 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 'Zeilennummer
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 [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 [DocId](#)** ([out, retval] BSTR \*pVal)  
**HRESULT [FileObjects](#)** ([out, retval] [IFileObjects](#) \*\*pVal)  
**HRESULT [GetFileObjectByID](#)** ([in] long lObjectID,[out, retval] [IFileObject](#) \*\*ppFileObject)  
**HRESULT [ObjectId](#)** ([out, retval] BSTR \*pVal)

### Dokumentation der Elementfunktionen

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

**Beschreibung:** Diese Methode liefert eine Liste aller Dokumente der Akte.

**Rückgabewerte:**

*IFileObjects:* Liste aller Dokumente

**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 Dokuments

*long*: Dokumenttyp

*VARIANT\_BOOL*: gibt an, ob das Objekt in den Arbeitsbereich der Akte gehört

*DocLocationEnum*: 1 = Dokument bereits zugeordnet 2 = Dokument liegt in der WF-Ablage

**Rückgabewerte:**

*VARIANT\_BOOL*: TRUE = Dokument hinzugefügt, andernfalls FALSE

**HRESULT DeleteFileObject ([in] long *lObjectID*, [out, retval] VARIANT\_BOOL \* *pbResult*)**

**Beschreibung:** Diese Methode löscht ein Dokument aus der Akte.

**Parameter:**

*long*: ID des Dokuments

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

**Beschreibung:** Veraltete Version. Bitte nicht mehr verwenden.

**HRESULT GetFileObjectByID ([in] long *lObjectID*, [out, retval] IFileObject \*\* *ppFileObject*)**

**Beschreibung:** Diese Methode liefert ein Dokument anhand seiner Dokument-Id.

**Parameter:**

*long*: Dokument-ID

**Rückgabewerte:**

*IFileObject*: Dokumentenobjekt

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

**Beschreibung:** Veraltete Version. Bitte nicht mehr verwenden.

## IWFItem Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt einen startbaren Vorgang (wird im enaio® - Workflow Client in Registerkarte "Startbare Vorgänge" angezeigt). Zugriff auf diese Schnittstelle über:

- [IWFCClient::WFItems\(\)](#)

### Öffentliche Methoden

**HRESULT [ID](#)** ([out, retval] BSTR \*pVal)  
**HRESULT [InfoText](#)** ([out, retval] BSTR \*pVal)  
**HRESULT [InputVariables](#)** ([out, retval] [IWFVariables](#) \*\*ppVariables)  
**HRESULT [Name](#)** ([out, retval] BSTR \*pVal)  
**HRESULT [StartProcess](#)** ([in] [IFileObjects](#) \*pFileObjects, [in] [IWFVariables](#) \*pVariables, [out] VARIANT \*pvarNewProcessID, [out] VARIANT \*pvarErrorText, [out, retval] long \*plResult)  
**HRESULT [WorkflowID](#)** ([out, retval] BSTR \*pVal)

### Dokumentation der Elementfunktionen

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

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

**Rückgabewerte:**

*BSTR:* Name des Vorgangs

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

**Beschreibung:** Diese Methode liefert die ID der Workflowfamilie aus dem enaio® - Workflow Editor.

**Rückgabewerte:**

*BSTR:* ID der Workflowfamilie

**HRESULT StartProcess** ([in] [IFileObjects](#) \* pFileObjects, [in] [IWFVariables](#) \* pVariables, [out] VARIANT \* pvarNewProcessID, [out] VARIANT \* pvarErrorText, [out, retval] long \* plResult)

**Beschreibung:** Diese Methode startet einen Vorgang (Workflow-Prozess). Der Methode kann eine Liste von enaio®-Dokumenten (werden in die WF-Akte eingefügt) und eine Liste aller Workflowvariablen übergeben werden.

**Parameter:**

*IFileObjects:* Liste von enaio®-Dokumenten

*IWFVariables*: Liste aller Workflowvariablen

*VARIANT*: hier wird die ID (Instanid) des gestarteten Vorgangs zurückgegeben.

*VARIANT*: sollte das Starten fehlschlagen, wird hier der Fehlertext zurückgegeben.

#### Rückgabewerte:

*long*: 0 = Vorgang gestartet, Ungleich 0 = Fehler aufgetreten, der Fehlertext ist in *pvarErrorText* enthalten

#### Beispiel:

```
'Objekte erstellen
Set oWFFiles = CreateObject("oxwfcInt.fileobjects") 'Liste von Dokumenten
Set oWFFile = CreateObject("oxwfcInt.fileobject") 'neues Dokument

'Dokument initialisieren
oWFFile.Id = 124
oWFFile.ObjectType = 19660800
oWFFile.IsWorkSpaceObject = true
oWFFile.Location = 2

'Dokument der Liste hinzufügen
oWFFiles.Add ObjectId, oWFFile

'Eingabevariablen holen
oWFItem.InputVariables(oVariables)

'Variable initialisieren
oVariables.Item("Name der Variable").Value = "Beispiel"

oWFItem.startprocess(oWFFiles, oVariables, varid, varerror)
```

Siehe auch: [IFileObject](#) , [IFileObjects](#)

### **HRESULT InputVariables ([out, retval] [IWFVariables](#) \*\* *ppVariables*)**

**Beschreibung:** Diese Methode liefert eine Liste aller Eingabeparameter des Vorganges.

**Rückgabewerte:**

*IWFVariables*: Liste aller Eingabeparameter

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

**Beschreibung:** Diese Methode liefert die Beschreibung die im enaio® - Workflow Editor hinterlegt wurde.

**Rückgabewerte:**

*BSTR*: Beschreibung zum Modell

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

**Beschreibung:** Diese Methode liefert eindeutige ID des Workflowmodells.

**Rückgabewerte:**

*BSTR*: Id des Workflowmodells

## IWFItems Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Liste aller startbaren Vorgänge für den aktuell angemeldeten Benutzer.  
Zugriff auf diese Schnittstelle über:

- [IWFCClient::WFItems\(\)](#)

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



## IWFListColumn Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Spalte eines Maskenfeldes in Tabellenform.

### Öffentliche Methoden

**HRESULT [DataType](#)** ([out, retval] BSTR \*pVal)  
**HRESULT [DBFieldName](#)** ([out, retval] BSTR \*pVal)  
**HRESULT [GetStringProperty](#)** ([in] BSTR strPropertyName,[in] BSTR strDefault,[out, retval] BSTR \*pbstrProperty)  
**HRESULT [Name](#)** ([out, retval] BSTR \*pVal)  
**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

ListColumns = wfmaskfields.item(MaskfieldName).WFListColumns

Set Column = ListColumns.item(MyPosition)

Value = Column.GetStringProperty(PropertyName, MyDefaultValue)
```

**Siehe auch:**

[SetStringProperty\(\)](#)

## **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:**

*BSTR*: Name der Eigenschaft  
*BSTR*: Wert der Eigenschaft

### **Rückgabewerte:**

*VARIANT\_BOOL*: TRUE, kein Fehler, ansonsten FALSE

**Beispiel:** Ein Wert (MyNewValue) 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 erstellen
Set Column = Maskfield.WFListColumns.item(MyPosition) 'Spaltenobjekt erstellen

count = 0

Do 'ersten freien Listeneintrag suchen

    PropertyName = "Zeile" & count 'Eigenschaftsname zusammensetzen

    Value = Column.GetStringProperty(PropertyName, MyDefaultValue) 'Listeneinträge
                                                                holen

    If Value = "free entry" Then 'freien Listeneintrag gefunden

        Result = Column.SetStringProperty(PropertyName, MyNewValue) 'neuer Eintrag
        Exit Do

    End If

    count = count + 1

Loop
```

Siehe auch: [GetStringProperty\(\)](#)

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

**Beschreibung:** Diese Methode liefert den Datenbankfeldnamen der Spalte.

**Rückgabewerte:**

*BSTR*: Datenbankfeldnamen der Spalte

## IWFListColumns Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Liste aller Spalten eines Maskenfeldes in Tabellenform. 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 Feld in Eingabemasken. In Client-Event erhält man ein Objekt folgendermaßen:

```
Dim MaskfieldName `Name des Maskenfelds`  
  
Set Maskfield = wfmaskfields.item(MaskfieldName)
```

Ein Maskenfeld-Objekt wird nur über den Namen des Feldes, wie er auf der Eingabemaske definiert wurde, angesprochen.

### Öffentliche Methoden

**HRESULT [CtrlPages](#) ([out, retval] [IWFCtrlPages](#)\* \*pVal)**  
**HRESULT [DataType](#) ([out, retval] BSTR \*pVal)**  
**HRESULT [Enabled](#) ([in] VARIANT\_BOOL newVal)**  
**HRESULT [Enabled](#) ([out, retval] VARIANT\_BOOL \*pVal)**  
**HRESULT [ID](#) ([out, retval] BSTR \*pVal)**  
**HRESULT [Name](#) ([out, retval] BSTR \*pVal)**  
**HRESULT [SectionName](#) ([out, retval] BSTR \*pVal)**  
**HRESULT [Visible](#) ([in] VARIANT\_BOOL newVal)**  
**HRESULT [Visible](#) ([out, retval] VARIANT\_BOOL \*pVal)**  
**HRESULT [WFFieldValues](#) ([out, retval] [IWFFieldValues](#) \*\*ppValues)**  
**HRESULT [WFListColumns](#) ([out, retval] [IWFListColumns](#) \*\*ppListColumns)**

### Dokumentation der Elementfunktionen

**HRESULT [WFFieldValues](#) ([out, retval] [IWFFieldValues](#) \*\* ppValues)**

**Beschreibung:** Diese Methode erlaubt den Zugriff auf die Auswahlliste, die zu einem Textfeld hinterlegt wurde.

**Rückgabewerte:**

*IWFFieldValues*: Eigenschaften

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

**Beschreibung:** Diese Methode liefert den Namen des Maskenfeldes, wie er auf der Eingabemaske definiert ist.

**Rückgabewerte:**

*BSTR*: Name des Maskenfeldes

**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)

**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

**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

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

### **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

### **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

### **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

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

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

**Rückgabewerte:**

*BSTR*: MaskenfeldId

### **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)
```

## IWFOrganisation Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Organisation. Zugriff auf diese Schnittstelle über:

- [IWFCClient::WFOrganisations\(\)](#)

### Öffentliche Methoden

**HRESULT [Active](#) ([out, retval] long \*pVal)**  
**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 Organisation.  
**Rückgabewerte:**  
*BSTR:* OrganisationsId

**HRESULT Name ([out, retval] BSTR \* pVal)**  
**Beschreibung:** Diese Methode liefert den Namen der Organisation.  
**Rückgabewerte:**  
*BSTR:* Organisationsname

**HRESULT Active ([out, retval] long \* pVal)**  
**Beschreibung:** Diese Methode liefert den Namen der Organisation.  
**Rückgabewerte:**  
*BSTR:* Organisationsname

## IWFOrganisations Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Liste aller Organisationen. Zugriff auf diese Schnittstelle über:

- [IWFCClient::WFOrganisations\(\)](#)

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

## IWFScript Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt ein VB-Skript, welches zu einem Event hinterlegt wurde.

Siehe auch: [IWFEvent](#)

### Öffentliche Methoden

**HRESULT [Code](#) ([out, retval] BSTR \*pbstrVal)**

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

**HRESULT [Type](#) ([out, retval] long \*plVal)**

### Dokumentation der Elementfunktionen

**HRESULT [Code](#) ([out, retval] BSTR \* *pbstrVal*)**

**Beschreibung:** Diese Methode liefert den im enaio® - Workflow Editor hinterlegten Skript-Code.

**Rückgabewerte:**

*BSTR*: ID der Aktivität

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

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

**Rückgabewerte:**

*BSTR*: ID des Skripts

**HRESULT [Type](#) ([out, retval] long \* *plVal*)**

**Beschreibung:** Diese Methode liefert den Typ des Skripts

**Rückgabewerte:**

*long*: Typ des Skripts

**Hinweis:** mögliche Werte für den Typ

- 1 = Event-Skript
- 2 = Global Server-Skript
- 3 = Global Client-Skript

## IWFVariable Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Workflowvariable. Da Variablen verschachtelt sein können, besitzt diese Schnittstelle 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 [FieldName](#)** ([out, retval] BSTR \*pVal)  
**HRESULT [ID](#)** ([out, retval] BSTR \*pVal)  
**HRESULT [InitialValue](#)** ([out, retval] BSTR \*pVal)  
**HRESULT [InputMode](#)** ([out, retval] long \*pVal)  
**HRESULT [Name](#)** ([out, retval] BSTR \*pVal)  
**HRESULT [Value](#)** ([in] BSTR newVal)  
**HRESULT [Value](#)** ([out, retval] BSTR \*pVal)  
**HRESULT [VarType](#)** ([out, retval] VarTypeEnum \*pVal)  
**HRESULT [VarDataType](#)** ([out, retval] VarTypeEnum \*pVal)

### Listenmethoden

Diese Methoden können nur bei Workflowvariablen in Listenform benutzt werden.

**HRESULT [AddListItem](#)** ([in] [IWFVariable](#) \*pListItem)  
**HRESULT [Count](#)** ([out, retval] long \*pVal)  
**HRESULT [CreateListItem](#)** ([out, retval] [IWFVariable](#) \*\*ppVariable)  
**HRESULT [Item](#)** ([in] VARIANT varItem, [out, retval] [IWFVariable](#) \*\*pVal)  
**HRESULT [ListType](#)** ([out, retval] ListTypeEnum \*pVal)  
**HRESULT [RemoveListItem](#)** ([in] VARIANT varItem)  
**HRESULT [Selected](#)** ([in] VARIANT\_BOOL newVal)  
**HRESULT [Selected](#)** ([out, retval] VARIANT\_BOOL \*pVal)  
**HRESULT [ClearList](#)**()  
**HRESULT [TypeID](#)** ([out, retval] BSTR \*pVal)



## Dokumentation der Elementfunktionen

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

**Beschreibung:** Diese Methode liefert ein Listenelement.

**Rückgabewerte:**

*IWFVariable*: Listenelement

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

**Beschreibung:** Diese Methode liefert die Anzahl der Listenelemente.

**Rückgabewerte:**

*long*: Anzahl

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

**Beschreibung:** Diese Methode liefert den Namen einer Variable.

**Rückgabewerte:**

*BSTR*: Name der Variable

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

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

**Rückgabewerte:**

*BSTR*: Variablenid

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

**Beschreibung:** Diese Methode liefert den Wert der Variable.

**Rückgabewerte:**

*BSTR*: Wert der Variable

### **HRESULT Value ([in] BSTR newVal)**

**Beschreibung:** Diese Methode setzt den Wert der Variable.

**Parameter:**

*BSTR*: Wert der Variable

### **HRESULT VarType ([out, retval] VarTypeEnum \* pVal)**

**Beschreibung:** Diese Methode liefert den Variablentyp.

**Rückgabewerte:**

*VarTypeEnum*: Typ

**Hinweis:** Mögliche Werte

- 0 = normale Stringvariable
- 1 = die Variable ist eine Liste
- 2 = die Variable ist ein Record

### **HRESULT VarDataType ([out, retval] VarDataTypeEnum \* pVal)**

**Beschreibung:** Diese Methode liefert den Variablentypen.

**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

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

**Beschreibung:** Diese Methode gibt Auskunft über die 'Beschreibbarkeit' der Variable.

**Rückgabewerte:**

*long*: Zugriffstyp

**Hinweis:** Mögliche Werte

- 1 = nur lesender Zugriff zulässig
- 2 = nur schreibender Zugriff
- 3 = lesender und schreibender Zugriff erlaubt

### **HRESULT CreateListItem ([out, retval] [IWFVariable](#) \*\* ppVariable)**

**Beschreibung:** Diese Methode erzeugt ein neues Objekt zum Einfügen in die Liste. Das neue Objekt kann mit AddListItem hinzugefügt werden.

**Rückgabewerte:**

*IWFVariable*: 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] [IWFVariable](#) \* pListItem)**

**Beschreibung:** Diese Methode fügt ein neues Objekt in die Liste aller Workflowvariablen ein.

**Parameter:**

*IWFVariable*: neues Element

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

### **HRESULT RemoveListItem ([in] VARIANT varItem)**

**Beschreibung:** Diese Methode löscht ein Listenelement.

**Parameter:**

*VARIANT*: Index des Listenelements

### **HRESULT ListType ([out, retval] ListTypeEnum \* pVal)**

**Beschreibung:** Diese Methode liefert den Listentyp.

**Rückgabewerte:**

*ListTypeEnum*: Listentyp

**Hinweis:** Mögliche Werte

- `ltUSERDEFINED` = 0,
- `ltPROCESSLIST` = 1,
- `ltUNKNOWN` = 100

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

**Beschreibung:** Diese Methode gibt an, ob ein Listenelement selektiert/markiert ist.

**Rückgabewerte:**

*VARIANT\_BOOL*: TRUE = Listenelement markiert

### **HRESULT Selected ([in] VARIANT\_BOOL newVal)**

**Beschreibung:** Diese Methode selektiert/markiert ein Listenelement.

**Parameter:**

*VARIANT\_BOOL*: TRUE = Listenelement markiert

### **HRESULT ClearList()**

**Beschreibung:** Diese Methode löscht die Liste.

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

**Beschreibung:** Liefert die Id des Datentyps der Listenelemente.

**Rückgabewert:**

*BSTR*: Id des in der Liste enthaltenen Datentyps

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

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

**Rückgabewerte:**

*BSTR*: Defaultwert

## IWFVariables Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Liste aller verfügbaren Workflowvariablen. Diese Liste ist in Client-Events unter dem Namen 'WFVariables' verfügbar.

Elemente der Liste: [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

## IWFWorkItem Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt einen startbaren Vorgangsschritt im Eingangskorb. Zugriff auf diese Schnittstelle über:

- [IWFCClient::WFWorkItems\(\)](#)

### Öffentliche Methoden

**HRESULT [ActivityID](#)** ([out, retval] BSTR \*pVal)  
**HRESULT [ActivityName](#)** ([out, retval] BSTR \*pVal)  
**HRESULT [Cancel](#)** ([out] VARIANT \*pvarErrorString,[out, retval] long \*lpResult)  
**HRESULT [CreationTime](#)** ([out, retval] long \*pVal)  
**HRESULT [Execute](#)** ([out] VARIANT \*pvarResult,[out] VARIANT \*pvarErrorString,[out, retval] [IActiveWorkItem](#) \*\*ppActWorkItem)  
**HRESULT [GetVarName](#)** ([in] long lVarNum,[out, retval] BSTR \*pbstrVarName)  
**HRESULT [GetVarValue](#)** ([in] long lVarNum,[out, retval] BSTR \*pbstrVarValue)  
**HRESULT [GetVarDataType](#)**([in] long lVarNum,[out, retval] VarDataType \*pVal)  
**HRESULT [ID](#)** ([out, retval] BSTR \*pVal)  
**HRESULT [OverTime](#)** ([out, retval] VARIANT\_BOOL \*pVal)  
**HRESULT [ProcessID](#)** ([out, retval] BSTR \*pVal)  
**HRESULT [ProcessName](#)** ([out, retval] BSTR \*pVal)  
**HRESULT [State](#)** ([in] WorkItemStateEnum newVal)  
**HRESULT [State](#)** ([out, retval] WorkItemStateEnum \*pVal)  
**HRESULT [VarCount](#)** ([out, retval] long \*pVal)  
**HRESULT [WarningTime](#)** ([out, retval] long \*pVal)  
**HRESULT [WorkflowID](#)** ([out, retval] BSTR \*pVal)  
**HRESULT [WorkflowVersion](#)**([out, retval] long \*pVal)  
**HRESULT [ProcessSubject](#)**([out, retval] BSTR \* pVal)  
**HRESULT [ObjectId](#)** ([out, retval] long \*pVal)

### Dokumentation der Elementfunktionen

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

**Beschreibung:** Diese Methode liefert den Prozessnamen.

**Rückgabewerte:**

*BSTR*: Prozessname

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

**Beschreibung:** Diese Methode liefert die eindeutige ID des Vorgangsschrittes.

**Rückgabewerte:**

*BSTR*: Instanzid der Aktivität

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

**Beschreibung:** Diese Methode liefert den Status des Vorgangsschrittes.

**Rückgabewerte:**

*WorkItemStateEnum*: Status

**Hinweis:** mögliche Werte für den Status

- 0 = wsInactive
- 1 = wsInitiated
- 2 = wsRunning
- 4 = wsSuspended
- 16 = wsTerminated
- 32 = wsCompleted
- 40 = wsInUse

### **HRESULT State ([in] WorkItemStateEnum newVal)**

**Beschreibung:** Diese Methode setzt den Status des Vorgangsschrittes.

**Rückgabewerte:**

*WorkItemStateEnum*: Status

**Hinweis:** mögliche Werte für den Status

- 0 = wsInactive
- 1 = wsInitiated
- 2 = wsRunning
- 4 = wsSuspended
- 16 = wsTerminated
- 32 = wsCompleted
- 40 = wsInUse



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

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

**Rückgabewerte:**

*BSTR*: Name

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

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

**Rückgabewerte:**

*BSTR*: ProzessId

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

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

**Rückgabewerte:**

*BSTR*: Id der Aktivität aus dem Workflowmodell

### **HRESULT Execute ([out] VARIANT \* pvarResult, [out] VARIANT \* pvarErrorString, [out, retval] IActiveWorkItem \*\* ppActWorkItem)**

**Beschreibung:** Diese Methode startet einen Vorgangsschritt. Ein durch einen Benutzer gestarteter Vorgang ist für andere nicht mehr sichtbar. Er wird aus der Liste "IWFWorkItems" entfernt und in ein Objekt der Schnittstelle "IActiveWorkItem" überführt.

**Parameter:**

*VARIANT*: hier wird 0 zurückgegeben, wenn kein Fehler aufgetreten ist.

*VARIANT*: hier wird im Falle eines Fehlers (lpResult <> 0) der Fehlertext zurückgegeben.

**Rückgabewerte:**

*IActiveWorkItem*: gestarteter Vorgangsschritt

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

**Beschreibung:** Diese Methode liefert die Mahnfrist in Form eines Zeitstempels.

**Rückgabewerte:**

*long*: Zeitstempel

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

**Beschreibung:** Diese Methode liefert die Anzahl der Workflowvariablen die im Eingangskorb angezeigt werden sollen.

**Rückgabewerte:**

*long*: Anzahl der Workflowvariablen

Siehe auch:

[GetVarName\(\)](#), [GetVarValue\(\)](#)

### **HRESULT GetVarName ([in] long IVarNum, [out, retval] BSTR \* pbstrVarName)**

**Beschreibung:** Diese Methode liefert den Namen einer Workflowvariablen die im Eingangskorb angezeigt werden sollen.

**Parameter:**

*long*: Index der geforderten Variablen.

**Rückgabewerte:**

*BSTR*: Name der Variable

Siehe auch:

[GetVarValue\(\)](#), [VarCount\(\)](#)

### **HRESULT GetVarValue ([in] long IVarNum, [out, retval] BSTR \* pbstrVarValue)**

**Beschreibung:** Diese Methode liefert den Wert einer Workflowvariablen die im Eingangskorb angezeigt werden sollen.

**Parameter:**

*long*: Index der Workflowvariablen

**Rückgabewerte:**

*BSTR*: Wert der Variable

Siehe auch:

[GetVarName\(\)](#), [VarCount\(\)](#)

### **HRESULT GetVarDataType ([in] long IVarNum, [out, retval] VarDataType \*pVal)**

**Beschreibung:** Diese Methode liefert den Datentyp einer Workflowvariablen die im Eingangskorb angezeigt werden sollen.

**Parameter:**

*long*: Index der Workflowvariablen

**Rückgabewerte:**

*BSTR*: Datentyp der Variable

Siehe auch:

[GetVarName\(\)](#), [VarCount\(\)](#), [IWFVariable::VarDataType\(\)](#)

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

**Beschreibung:** Diese Methode gibt an, wann der Vorgang gestartet wurde.

**Rückgabewerte:**

*long*: Startzeit

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

**Beschreibung:** Diese Methode entpersonalisiert einen Vorgangsschritt. Der Vorgangsschritt ist dann keinem Benutzer mehr zugeordnet und kann wieder durch einen anderen Benutzer personalisiert werden.

**Parameter:**

*VARIANT*: hier wird ggf. der Fehlertext zurückgeliefert

**Rückgabewerte:**

*long*: 0 = kein Fehler aufgetreten

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

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

**Rückgabewerte:**

*BSTR*: Id des Workflowmodells

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

**Beschreibung:** Diese Methode liefert die Version des Workflowmodells.

**Rückgabewerte:**

*LONG*: Version des Workflowmodells

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

**Beschreibung:** Diese Methode gibt an, ob die Mahnfrist überschritten wurde.

**Rückgabewerte:**

*VARIANT\_BOOL*: TRUE = Mahnfrist überschritten

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

**Beschreibung:** Diese Methode gibt den Inhalt des Betrefffeldes wieder.

**Rückgabewerte:**

*BSTR*: Inhalt des Betrefffeldes

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

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

**Rückgabewerte:**

*LONG*: Version des Workflowmodells

## IWFWorkItems Schnittstellenreferenz

### Ausführliche Beschreibung

Diese Schnittstelle beschreibt eine Liste aller Vorgangsschritte im Eingangskorb. Zugriff auf diese Schnittstelle über:

- [IWFCClient::WFWorkItems](#) ()

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

## 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

### Anmeldung am enaio® und Verbindung zum Workflow (VBS)

```
Set EMCSsystem = CreateObject("axcom.assystem")
Set Workflow = CreateObject("oxwfcInt.wfclient")

connect = EMCSsystem.Login("AVG", "Name der Api", _
                           "Benutzer@Passwort", _
                           TRUE, FALSE, TRUE, FALSE)

WorkflowConnect = Workflow.connect

If WorkflowConnect = TRUE then

    'Verbindung besteht

Else
    'Verbindung konnte nicht erstellt werden
    msgbox "Connect schlug fehl"
End If
```

## Verwendung von Serverjobs und Start eines Workflows aus axadmin.exe

Dieses Beispielskript wird nach dem Import eines Word-Dokumentes in den Schrank Pressearchiv gestartet. Im Skript ist die Id des importierten Dokumentes bekannt. Mit Hilfe dieser Id wird eine DMS-Recherche durchgeführt und aus den Indexdaten der Autor ermittelt. Für den Autor wird dann ein Workflowprozess gestartet (Im Workflowprozess muss dann im Startevent der Autor als Teilnehmer gesetzt werden).

```
'Dokumentid ermitteln
Strarray = Split(arrObjects(0), "/")
Objarray = Split(Strarray(2), "#")
IDarray = Split(Objarray(1), ":")
objectid = IDarray(0)

'für das Absetzen von Serverjobs wird die Schnittstelle oxmljsc verwendet
Set goxmljsc = CreateObject("oxmljsc.oxmljsc")

'Objekt mit IP-Adresse und Port des Servers initialisieren
Result = goxmljsc.Init(Errors, "IPAddress=Server;TCPPort=4000")

If Result = 0 Then
    'Verbindung zum Server herstellen
    Result = goxmljsc.Connect(Errors)

    If Result = 0 Then
        'Daten der Workstation ermitteln
        Result = goxmljsc.GetParameter(Errors, "StationGUID",
varStationGUID)
        Result = goxmljsc.GetParameter(Errors, "StationIP",
varStationIP)
        Result = goxmljsc.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 = goxmljsc.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",  
"BP1157531646352712075471225323")  
        Result = goxmljsc.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>" & objectid & "</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 = goxmljsc.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  
  
                Set wfitems = wfclient.wfitems  
  
                'Workflowmodell ermitteln  
                For ix = 0 To wfitems.count - 1  
                    Set wfitem = wfitems.item(ix)
```

```
        If wfitem.name = "Test-Modell" Then
            Set Variables = wfitem.InputVariables

            Set varname = Variables.item(0)
            'Name des Autor dem Workflow übergeben,
            'diesem dann den Workflow zuzuweisen
            varname.value = name

            iRet = wfitem.startprocess(Nothing,
                                      Variables, varid,
varerror)

            Exit For
        End If
    Next

    Exit For
End If
Next

'Logout durchführen
JobName = "krn.SessionLogout"
ParNmIn = Array("Flags")
ParValIn = Array(0)
Result = goxmljsc.DoJob(Errors, JobName, ParNmIn, ParValIn,
                        FileNmIn, ParNmOut, ParNmOut,
FileNmOut)

'Verbindung zum Server beenden
goxmljsc.Disconnect (Errors)
End If

goxmljsc.Finalize (Errors)
End If
```

## Programmierbeispiel in Visual Basic

```
Dim workitems As WFWorkItems
Dim wfworkitem As wfworkitem
Dim WithEvents wfclient As OXWFCLNTLib.wfclient
Private Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)

'-----
--
Private Sub CreateConnectionClick() 'Anmeldung am Workflowsystem

    Set wfclient = CreateObject("oxwfclnt.wfclient")

    If wfclient Is Nothing Then
        MsgBox "wfclient not created"
    Else

        'UseRapidTranslation wird in VB nicht unterstützt
        wfclient.UseRapidTranslation = False

        'mit Workflowsystem verbinden
        bConnected = wfclient.Connect

        If Not bConnected = True Then
            MsgBox "not connected"
        End If

    End If

End Sub

Private Sub RunningActivitiesClick() 'Laufende Vorgänge anzeigen

    Set runningactivities = wfclient.RunningActivities

    If Not runningactivities Is Nothing Then
        For i = 0 To runningactivities.Count - 1
            Set runningactivity = runningactivities(i)
            MsgBox runningactivity.Name
        Next
    End If

End Sub

'-----
--
```

```
Private Sub GetWorkitemsClick() 'Eingangskorb für angemeldeten Benutzer anzeigen

    'Eingangskorb von Server neu anfordern
    wfclient.RefreshWorkItemList = True

    Set workitems = wfclient.WFWorkItems
    If Not workitems Is Nothing Then
        'alle Vorgänge anzeigen
        For i = 0 To workitems.Count - 1
            Set workitem = workitems(i)
            MsgBox workitem.ProcessName
        Next
    End If

End Sub

-----
Private Sub StartProcessClick() 'Prozess starten

    'alle startbaren Vorgänge für angemeldeten Benutzer ermitteln
    Set workflows = wfclient.WFItems

    'nach Vorgang "Test" suchen
    Set workflow = Nothing
    For i = 0 To workflows.Count - 1
        If workflows(i).Name = "Test" Then
            Set workflow = workflows(i)
            Exit For
        End If
    Next

    If workflow Is Nothing Then
        MsgBox "No workflow model found"
    Else

        'Startparameter-Objekt für den Prozess erstellen
        Set variables = workflow.InputVariables

        'jetzt können Variablen gesetzt werden

        'Dokumentenlisten-Objekt für Workflowakte erstellen
        Set file = CreateObject("oxwfclnt.FileObjects")

        'jetzt können Dokumente gesetzt werden

        'Prozess starten
        varResult = workflow.StartProcess(file, variables, varid, varerror)

    End If

End Sub
```

```
Private Sub ExecuteCompleteClick() 'Vorgangsschritt starten und
Weiterleiten

    'Eingangskorb neu anfordern
    wfclient.RefreshWorkItemList = True

    'Eingangskorb erstellen
    Set workitems = wfclient.WFWorkItems

    If Not workitems Is Nothing Then

        If workitems.Count = 0 Then
            MsgBox "workitem list is empty"
        Else

            Set workitem = Nothing

            'nach Vorgang mit Namen 'Test' suchen
            For i = 0 To workitems.Count - 1

                If Left(workitems(i).ProcessName, 4) = "Test" Then
                    Set workitem = workitems(i)
                    Exit For
                End If

            Next

            If Not workitem Is Nothing Then

                MsgBox "Execute workitem (" & workitem.ProcessName & ")"

                'Vorgangsschritt starten
                Set activeworkitem = workitem.Execute(varResult, varerror)

                If Not activeworkitem Is Nothing Then

                    'hier kann jetzt der Vorgangsschritt bearbeitet werden

                    'Beispiel: die Workflowvariablen zum Vorgangsschritt
                    'werden ermittelt
                    Set variables = activeworkitem.WFVariables

                    'Vorgangsschritt weiterleiten
                    varResult = activeworkitem.Complete(varerror)
```

```
                If varResult <> 0 Then
                    MsgBox "error occured while completing workitem:" &
varerror
                End If
            Else
                MsgBox "can not execute workitem: " & varerror
            End If

            Else
                MsgBox "not workitem to process"
            End If
        End If
    End If
End Sub
```