



DOCUMENTS

SOAP

Schnittstellen-Dokumentation

DOCUMENTS 5

© Copyright 2016 otrs software AG. Alle Rechte vorbehalten.

Weitergabe und Vervielfältigung dieser Publikation oder von Teilen daraus sind, zu welchem Zweck und in welcher Form auch immer, ohne die ausdrückliche schriftliche Genehmigung durch die otrs software AG nicht gestattet. In dieser Publikation enthaltene Informationen können ohne vorherige Ankündigung geändert werden.

Alle in dieser Publikation aufgeführten Wort- und Bildmarken sind Eigentum der entsprechenden Hersteller.

Änderungen in der Software sind vorbehalten. Die in diesem Handbuch enthaltenen Informationen stellen keinerlei Verpflichtung seitens des Verkäufers dar.

Inhaltsverzeichnis

1.	Der DOCUMENTS SOAP Webservice	4
2.	Vom Service zum Projekt.....	9
2.1	WSDL	9
2.2	WSDL und Programmiersprache	10
2.3	WSDL-Tool und SOAP-Session	12
2.4	Documents- WSDL- Version	13
2.5	Arbeiten in einem Microsoft® Visual Studio 2005® Projekt	13
2.6	Beispielcode Konsolenanwendung VB	14
2.7	Erstellung eines gSOAP-Clients mit Visual C++ 2008®	16
2.8	Erstellung eines WCF-Clients mit Visual C# 2008®	21
3.	API	27
3.1	DOCUMENTS-Methoden	27
3.2	DOCUMENTS-Klassen	76
3.3	Weitere Funktionen.....	87
4.	HTTPS-Unterstützung (SSL/TLS)	89
4.1	Konfigurationen in der docsoaproxy.ini	89
4.2	Erstellen eines selbst signierten X.509-Zertifikats.....	90
4.3	HTTPS/SSL fähiger gSOAP-Client mit Visual C++ 2008®	90
4.4	Weitere HTTPS/SSL fähige Clients	92
5.	Anhang.....	93
5.1	Archivmappen-Key	93
5.1.1	EE.i:	93
5.1.2	EE.x:	94
5.1.3	EAS und EBIS:.....	95
5.2	Suchquellen-Id eines Archives.....	96
5.3	Archivzielbezeichner.....	96
5.4	Feldbezeichnerfür die Ausgabe von Mappenattributen in Reports/Suche.....	97
5.5	Bezeichner für persönliche Standardordner	97
5.6	Syntaxbeschreibung für Filterausdrücke	97
5.7	Schema für GetFilingPlanXML-Ausgabe	98
5.8	Client-Timeouts	99
6.	Index	101
7.	Abbildungsverzeichnis.....	106

1. Der DOCUMENTS SOAP Webservice

DOCUMENTS stellt einen Webservice bereit mit dessen Hilfe externe Anwendungen über eine **SOAP**-Schnittstelle auf **DOCUMENTS**-Mappen zugreifen und diese bearbeiten können.

Funktionsumfang

Zurzeit sind folgende Operationen auf verschiedenen Objektebenen möglich:

User

- Ein- und Ausloggen am **DOCUMENTS**-Proxy.
- Anmelden eines Benutzers unter einem anderen Account.
- Aktuellen Benutzer der Schnittstelle ermitteln.
- Benutzereigenschaften auslesen

Ordner

- Eingangsordner des Benutzers auslesen.
- Gesendet-Ordner des Benutzers auslesen.
- Ordner und Unterordner ermitteln (für Standardordner).
- Öffentliche Ordner auslesen.

Mappentypen

- Mappentypen ermitteln.
- Mappentypinformationenauslesen zu:
- Feldern, Archiven, Dokumentenregistern, Workflows

Mappen

- Anlegen, editieren und löschen.
- Mappen an neue Empfänger versenden (Parallel zur Info, Sequentiell).
- Dokumente hinzufügen, auslesen und löschen.
- Aufgaben auslesen.
- Monitor auslesen.
- Mappen archivieren.
- Wiedervorlagedatum setzen
- Eigenschaften und Autotexte auslesen

Suche

- Suchen in Mappentypen und Archiven durchführen.

Workflow

- Workflows ermitteln.
- Workflows auf Mappen starten und beenden.
- Benutzeraktionen auf Mappen auslesen und ausführen.
- Skripte auf Mappen starten.

Aktenpläne

- Aktenpläne ermitteln.
- XML-Darstellung des Aktenplans auslesen

WSDL

- WSDL- Download

Webservice und Proxy

Der **DOCUMENTS**-Webservice ist nicht vollständig in den **DOCUMENTS**-Server integriert. Er wird über einen speziellen Proxyserver („DOCUMENTS-Proxy“) bereitgestellt. Die programmierte Anwendung ist dabei ein Client für den Proxy, der seinerseits als Client gegenüber dem **DOCUMENTS**-Server auftritt. Der Proxy verwandelt die eingehenden SOAP-Anfragen in systemeigene Funktionsaufrufe, und er unterstützt den **DOCUMENTS**-Server beim Sitzungsmanagement für SOAP-Clients. Damit die Anwendung den Proxy erreichen kann, muss gegebenenfalls die Adresse angepasst werden, unter der die Kommunikation stattfindet (s.u. URL).

Um mit der vorliegenden API arbeiten zu können, müssen folgende Voraussetzungen gegeben sein:

Um mit der vorliegenden API arbeiten zu können, müssen folgende Voraussetzungen gegeben sein:

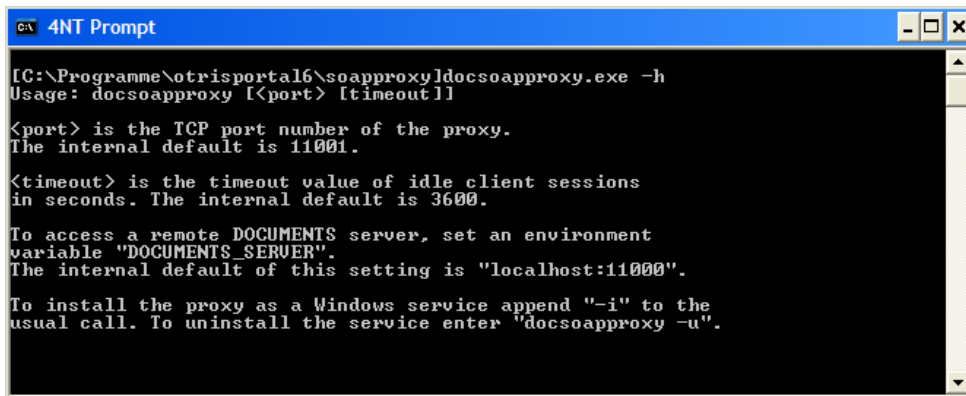
1. Ein installierter **DOCUMENTS**-Server
2. Eine Lizenzdatei mit eingeschlossener **DOCUMENTS**-API-Lizenz. Bei anderen Lizenzen kann der Service auf wenige Funktionen eingeschränkt oder auch komplett außer Funktion sein.
3. Installation des **DOCUMENTS**-Proxy

Der Proxy im Rahmen der **DOCUMENTS**-Installation im Unterverzeichnis `soapproxy` mitgeliefert.

Im Installationsverzeichnis des *Soaproxies* befindet sich auch eine aktuelle Dienstbeschreibungsdatei (WSDL-Datei). Sie ist für den Proxy und für die Programmierung von Clientanwendungen maßgeblich.

Ein Kommandozeilenaufbau des Proxies mit der Option `-h` listet weitere mögliche Kommandozeilenparameter auf.

Es können Vorgaben für Port und Timeout gemacht werden. Eine (De-)Installation als Windows-Dienst ist ebenfalls direkt über die Kommandozeile möglich. Die Verbindung zum **Documents**-Server kann anhand einer Umgebungsvariablen festgelegt werden.



```
4NT Prompt
[C:\Programme\otrisportal6\soaproxy\docsoaproxy.exe -h
Usage: docsoaproxy [<port> [<timeout>]]

<port> is the TCP port number of the proxy.
The internal default is 11001.

<timeout> is the timeout value of idle client sessions
in seconds. The internal default is 3600.

To access a remote DOCUMENTS server, set an environment
variable "DOCUMENTS_SERVER".
The internal default of this setting is "localhost:11000".

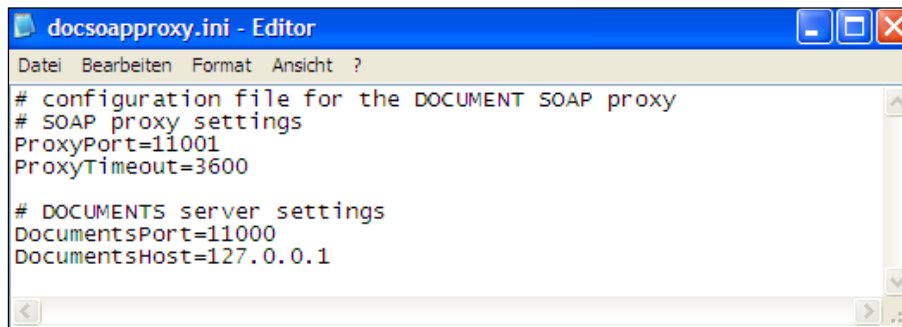
To install the proxy as a Windows service append "-i" to the
usual call. To uninstall the service enter "docsoaproxy -u".
```

Abb. 1: Kommandozeilenaufruf docsoaproxy -h

```
docsoaproxy 10123 3000
```

Aufruf mit Port 10123 und Timeout 3000

Die Proxy-Konfiguration kann auch über eine ini-Datei gesteuert werden, die in demselben Verzeichnis wie der Proxy selbst liegen muss. :



```
docsoaproxy.ini - Editor
Datei Bearbeiten Format Ansicht ?
# configuration file for the DOCUMENT SOAP proxy
# SOAP proxy settings
ProxyPort=11001
ProxyTimeout=3600

# DOCUMENTS server settings
DocumentsPort=11000
DocumentsHost=127.0.0.1
```

Abb. 2: docsoaproxy.ini als Konfigurationsdatei

Einstellungen in der Konfigurationsdatei überschreiben entsprechende Angaben in der Kommandozeile oder in einer Umgebungsvariablen.

- ProxyPort:
Der Port unter dem der Proxy die Clientanfragen entgegennimmt. Siehe auch unter: URL.

```
Voreinstellung: 11001
```

- ProxyTimeout:
Die Anzahl der Sekunden nach denen eine Session verfällt, wenn sie nicht weitergeführt oder durch ein Logout beendet wird.

```
Voreinstellung: 300
```

Das Sequenzdiagramm aus Abb. 3 veranschaulicht das Auftreten eines Proxy-Timeouts nach 3600s (siehe auch Client-Timeout).

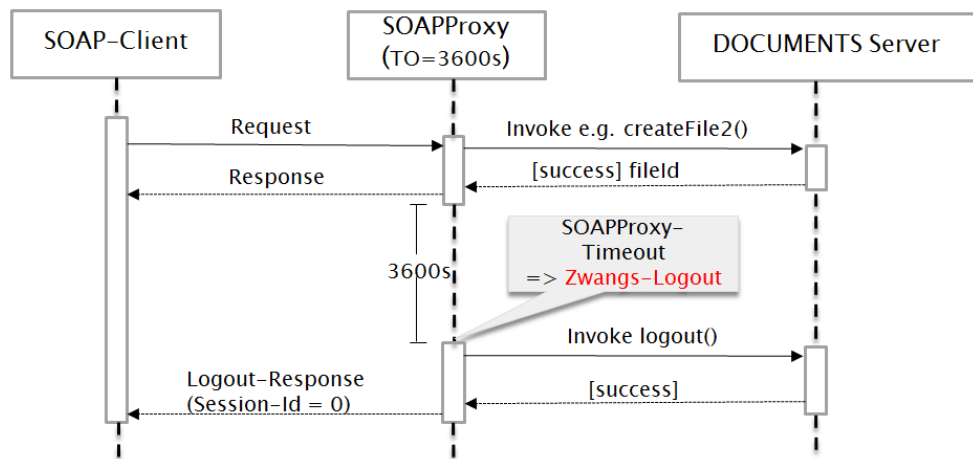


Abb. 3: Sequenzdiagramm eines Proxy-Timeouts

- DocumentsPort:
Der Port auf dem der **DOCUMENTS**-Server die Anfragen des Proxies entgegennimmt.

Default: 11000

- DocumentsHost:
Der Hostname, auf dem der **DOCUMENTS**-Server gestartet wurde.

Default: localhost

- SequenceSessionId:
Die ständig wechselnde Vergabe von *SessionIds* kann durch den Eintrag `SequenceSessionId=0` unterbunden werden (siehe WSDL-Tool und SOAP-Session). Aus Sicherheitsgründen wird dies nicht empfohlen.
- PortalServerEncoding:
Mit Version **ELC 3.60g/otris Portal 6.0g** kann für den Proxy das Encoding festgelegt werden, in dem die Daten des Servers vorliegen. Soweit nicht anders festgelegt, wird das *Systemencoding* verwendet und bei Start des Proxies angezeigt, z.B.: `German_Germany.1252`. Dabei erfolgt für Windows der Aufbau des Encodings in der Form:

[Language]_[Country].[CodePage]

Weitere Angaben dazu finden sich unter dem Stichwort `Language Identifiers and Locales` in der MSDN® von Microsoft®.

Kommt ein UTF-8 Portalserver (ab **DOCUMENTS 5** obligatorisch) zum Einsatz so muss unabhängig von Sprache und Land folgender Wert gesetzt werden:

PortalServerEncoding=UTF-8

Beachten Sie bitte, dass bei Verwendung eines Encodings, das vom Systemencoding abweicht, der SOAP-Client dies beachten muss und ggfs. sein Encoding anzupassen hat. Dies gilt auch für UTF-8 Encoding. Siehe u. *Codepage*

- BlobBasePath:
Ab der WSDL **DOCUMENTS-4.0.1870** ist es möglich, Blobs direkt aus dem Dateisystem hochzuladen. Diese Option legt ein Freigabeverzeichnis fest, in dem die Blobs sich befinden und auf das der SOAPProxy zugreifen kann. Der SOAP-Client übergibt dann

nicht mehr den Base64-kodierten Inhalt einer Datei, sondern ihren Pfadnamen relativ zu dem festgelegten Verzeichnis. Von dort nimmt sich der SOAPProxy die Datei und lädt sie zum **DOCUMENTS**-Server hoch.

- **SSL:**
Mit `SSL=1` wird SSL- oder TLS-verschlüsselte Kommunikation zwischen Client und SOAPProxy aktiviert. Diese Option ist verfügbar ab der WSDL DOCUMENTS-4.0.1870.
- **keyfile:**
Absoluter Pfad oder zum Installationsverzeichnis relativer Pfad zu einer Zertifikats-/Schlüssel-Datei für die SSL-Verschlüsselung. Die pem-Datei enthält den privaten RSA-Schlüssel und das Zertifikat des SOAPProxys, ggf. gefolgt von den Zwischenzertifikaten einer Zertifikatskette.
- **keypasswd:**
Das Kennwort zum Lesen des privaten Schlüssels in dem „keyfile“. Das Kennwort muss im Klartext angegeben werden.

Ab Version **ELC 3.60g** bzw. **otris Portal 6.0g** wird zusätzlich ein Proxy namens *docsoaproxy_log.exe* ausgeliefert, der die SOAP-Kommunikation protokolliert und somit das Debugging in der Kommunikationsschicht erleichtert. Siehe dazu auch Abschnitt *WSDL-Tool und SOAP-Session*.

2. Vom Service zum Projekt

2.1 WSDL

Definiert wird der angebotene Webservice über die **Web Service Definition Language (WSDL)**, die als Metasprache im XML-Format Austauschprotokolle vorgibt, mit denen Servermethoden von außen zugänglich gemacht werden können.

```
<!-- operation request element -->
<element name="getFileTypes">
  <complexType>
    <sequence>
      <element name="ignoreRight" type="xsd:boolean" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>
<!-- operation response element -->
<element name="getFileTypesResponse">
  <complexType>
    <sequence>
      <element name="filetype" type="DOCUMENTS:FileTypeShortDescr" minOccurs="0"
maxOccurs="unbounded"/>
    </sequence>
  </complexType>
</element>
```

Auszug aus der WSDL für die Methode „getFileTypes“

In dem obigen WSDL-Abschnitt wird beschrieben, wie ein Request zum Server abzusetzen ist. Es muss der Methode „getFileTypes“ genau ein Parameter „ignoreRight“ übergeben werden, worauf vom Server als Response (Antwort) eine unbestimmte Anzahl von DOCUMENTS:FileTypeShortDescr-Objektengeschickt werden. So wohl definiert und präzise die WSDL auch ausfällt, eine direkte Programmierung ist damit nicht oder nur sehr schwer möglich.

2.2 WSDL und Programmiersprache

Das Vorliegen einer Servicebeschreibung in Form einer WSDL ist also nur der erste Schritt hin zum Programmieren. Damit der bereitgestellte Dienst auch in Anspruch genommen werden kann, müssen aus der WSDL für die jeweilige Programmiersprache und IDE passende Strukturen und Klassen erzeugt werden. Die aktuelle WSDL wird im `soaproxy`-Verzeichnis bereitgestellt. Sie kann bei aktivem Proxy auch über `http://soaproxy:11001/?wsdl` geladen werden (siehe WSDL), beziehungsweise über `https://soaproxy:11001/?wsdl` für die HTTPS-Unterstützung ab WSDL DOCUMENTS-4.0.1870.

In Microsoft® Visual Studio 2005® wird diese Generierung durch das Programm „`wsdl.exe`“ geleistet, das über die `/language` Option eine Generierung in der Zielsprache ermöglicht. Ab Microsoft® Visual Studio 2008® Programm befindet sich in dem Windows-SDK, dass zur jeweiligen Visual Studio Version ausgeliefert wird (z.B. `C:\Programme\Microsoft SDKs\Windows\v6.0A\bin`).

```
"C:\Programme\Microsoft Visual Studio 8\SDK\v2.0\Bin\wsdl" /nologo /language:VB
/out:.\DOCUMENTS.vb /protocol:SOAP .\DOCUMENTS.wsdl
```

Aufruf zur Generierung der „DOCUMENTS.vb“ in Visual Basic aus der DOS-Shell im WSDL-Verzeichnis – `genWrapper.cmd`

Mit Vorlage der „`DOCUMENTS.vb`“ können nun in Visual Basic mit den aus der WSDL abgeleiteten Klassen sowohl Objekte der Schnittstelle erzeugt als auch Objektmethoden aufgerufen werden.

Aufgrund der gegebenen Abhängigkeit zieht eine Änderung in der WSDL eine Änderung im generierten Code der „`DOCUMENTS.vb`“ nach sich.

Zu beachten ist dabei, dass je nach Programmiersprache und eingesetztem WSDL-Tool der generierte Code anders ausfallen kann. Bei bestimmten Generierungswerkzeugen werden Objektansammlungen in einem Array bereitgestellt, während andere Tools diese in einer gesonderten Liste verwalten. Manche Tools kennen keinen ausgezeichneten `ReturnValue`, während andere den ersten Ausgabeparameter automatisch als `ReturnValue` der Funktion generieren.

Von der Programmiersprache sind natürlich auch die angebotenen Datentypen abhängig. Was als „`String`“ oder „`Integer`“ in der WSDL definiert wurde, wird in der Regel auch in der Zielsprache als „`String`“ oder „`Integer`“ angeboten. Dies kann aber bei einem „`ByteArray`“ in Base64 Codierung schon ganz anders aussehen. Bei Abweichungen in Datentypen und Signaturen sollten die WSDL und die aus ihr generierte **DOCUMENTS**-Datei miteinander verglichen werden.

So ergibt sich für das obige WSDL-Beispiel „*getFileTypes*“ in der generierten `DOCUMENTS.vb` folgende Implementierung:

```
Public Function getFileTypes(...ByVal ignoreRight As Boolean) As FileTypeShortDescr()  
    Dim results() As Object = Me.Invoke("getFileTypes", New Object() {ignoreRight})  
    Return CType(results(0),FileTypeShortDescr())  
End Function
```

Auszug aus der Documents.vb „getFileTypes“ – Einige Attribute wurden der Übersicht wegen ausgelassen.

Wichtig ist hier aber nicht nur die Implementierung dieser Methode, sondern ihr Aufruf aus dem Programm heraus, der sich geradezu trivial gestaltet.

```
Dim doc as New DOCUMENTS  
  
Dim ftsd() As FileTypeShortDescr  
ftsd = doc.getFileTypes(False)
```

Aufruf der Documents.vb „getFileTypes“ –Methode in einem Programm

Wie in dem Beispiel dargestellt, wird auch die Klasse `FileTypeShortDescr` mit all ihren in der WSDL festgelegten Mitgliedern generiert.

Für die Programmierung ergibt sich somit ein komfortables Arbeiten auf Basis von bereits fertig gestellten Klassen und Methoden. Ein direktes Arbeiten auf der darunter liegenden SOAP-Schicht ist gar nicht mehr notwendig.

Wenn der **DOCUMENTS**-Proxy nicht auf dem Rechner läuft, auf dem auch die Anwendung entwickelt wird, so muss nach der Instanziierung des **DOCUMENTS**-Objektes die URL, auf der der Proxy die Anfragen der programmierten Anwendung entgegennimmt, spezifiziert werden.

Die Default-Einstellung aus der WSDL „`http://localhost:11001`“ gilt, wenn Proxy und Anwendung auf demselben lokalen Rechner laufen.

Die URL setzt sich dabei aus dem Rechnernamen, unter dem dieser im Netz erreichbar ist, bzw. seiner IP-Adresse und dem Port zusammen.

```
Dim doc as New DOCUMENTS  
  
'Proxyadresse vergeben  
doc.Url = "http://proxyRechnerOderIpAdresse:11001"
```

Url unter der der DOCUMENTS-Proxy für die Applikation erreichbar ist.

2.3 WSDL-Tool und SOAP-Session

Die Kommunikation zwischen Client und Proxy wird über das SOAP-Protokoll abgewickelt. Dabei wird für jede einzelne Verbindung eine eindeutige `SessionId` vergeben, die im SOAP-Header mitgeführt wird und dafür sorgt, dass der Proxy die Antworten an den richtigen Client zurückschickt.

Der Kommunikationsablauf gestaltet sich dabei wie folgt:

```
Client A --> SessionId 0 --> Proxy

Client A meldet sich mit einem Login am Proxy an

Proxy -->SessionId 8186860391163 --> Client A
und erhält vom Proxy eine SessionId im SOAP-Header zurück

Client A -->SessionId 8186860391163 -->Proxy

Client A stellt eine weitere Anfrage an den Proxy und schickt im SOAP-Header die zuvor
erhaltene SessionId mit.

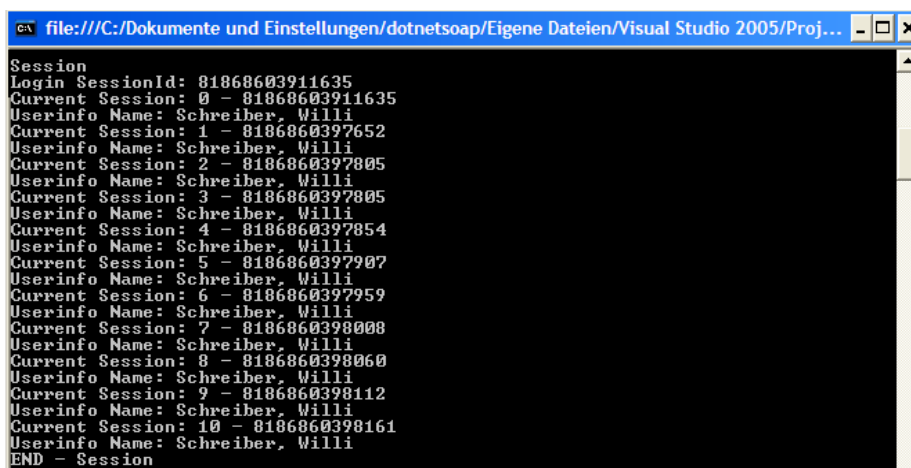
Proxy -->SessionId 8186860397652 -->Client A

Proxy erkennt, dass die Anfrage von Client A stammt, beantwortet diese und schickt dabei
eine neue (!) SessionId mit.
```

Session-Handling über SOAP

Bei jedem neuen Request eines Clients muss die zuvor in der Response erhaltene `SessionId` mitgeschickt werden. Erst diese Kette aus `SessionIds` kennzeichnet und identifiziert genau eine (1) Sitzung zwischen dem Client, Proxy und dem **DOCUMENTS**-Server. Dabei kann vom Proxy bei jeder Beantwortung des Requests eine neue `SessionId` zurückgeschickt werden. Sie muss sich aber nicht notwendigerweise ändern!

Dieser Mechanismus dient vorrangig dazu ein Session-Hijacking zu verhindern, bei dem ein Benutzer die Session eines bereits authentifizierten Benutzers an sich zieht. Im Folgenden wird eine Anmeldung durchgeführt und die Funktion `Userinfo` mehrfach hintereinander aufgerufen, bevor ein Logout erfolgt:



```
Session
Login SessionId: 81868603911635
Current Session: 0 - 81868603911635
Userinfo Name: Schreiber, Willi
Current Session: 1 - 8186860397652
Userinfo Name: Schreiber, Willi
Current Session: 2 - 8186860397805
Userinfo Name: Schreiber, Willi
Current Session: 3 - 8186860397805
Userinfo Name: Schreiber, Willi
Current Session: 4 - 8186860397854
Userinfo Name: Schreiber, Willi
Current Session: 5 - 8186860397907
Userinfo Name: Schreiber, Willi
Current Session: 6 - 8186860397959
Userinfo Name: Schreiber, Willi
Current Session: 7 - 8186860398008
Userinfo Name: Schreiber, Willi
Current Session: 8 - 8186860398060
Userinfo Name: Schreiber, Willi
Current Session: 9 - 8186860398112
Userinfo Name: Schreiber, Willi
Current Session: 10 - 8186860398161
Userinfo Name: Schreiber, Willi
END - Session
```

Je nach eingesetzten WSDL-Tool muss bei der Implementierung die `SessionId` dem SOAP-Header explizit hinzugefügt werden. Diese Einschränkung gilt nicht für Microsoft® Visual Studio® Projekte, hier geschieht das Setzen der `SessionId` automatisch im Hintergrund.

Die Vergabe wechselnder `SessionIds` kann in der ini-Datei des `Proxies` durch die Direktive `SequenceSessionId=0` unterbunden werden (siehe `SequenceSessionId`).

2.4 Documents- WSDL- Version

Die Versionsnummer der im Documents verwendeten WSDL befindet sich im `targetNamespace` der WSDL:

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="DOCUMENTS"
targetNamespace="http://xml.otris.de/ws/DOCUMENTS-4.0.1870.wsdl"
```

Version der verwendeten WSDL: **DOCUMENTS-4.0.1870.wsdl**

2.5 Arbeiten in einem Microsoft® Visual Studio 2005® Projekt

Für das Aufsetzen eines Microsoft® Visual Studio 2005® Projektes gilt:

1. Die Webservice Erweiterungen 3.0 von Microsoft müssen installiert sein:
 - Web Services Enhancements (WSE) 3.0 for Microsoft .NET
2. Es sind folgende Projektverweise hinzuzufügen:
 - System.Web.Services
 - System.Xml
3. Die WSDL-Datei sollte ebenso dem Projekt hinzugefügt werden:
 - Projektmappenexplorer->Kontextmenü->Hinzufügen->Vorhandenes Element->WSDL-Datei auswählen
4. Die Klassen aus der WSDL sind:
 - in die Zielprogrammiersprache zu generieren z.B. `DOCUMENTS.vb` s.o.
 - und dem Projekt hinzuzufügen:
 - Projektmappenexplorer->Kontextmenü->Hinzufügen->Vorhandenes Element->Datei auswählen

2.6 Beispielcode Konsolenanwendung VB

Im Folgenden gehen wir davon aus, dass in Microsoft® Visual Studio 2005® Visual Basic ein Projekt „docsample“ für eine Konsolenanwendung erstellt wurde und die oben geschilderten Verweise dem Projekt hinzugefügt wurden.

Fügen Sie dem Projekt eine Klasse „Sample“ hinzu:

```
Public Class Sample  
  
End Class
```

Fügen Sie weiter der Klasse „Sample“ eine „Shared Sub Main“-Methode hinzu:

```
Public Class Sample  
  
    Public Shared Sub Main()  
  
    End Sub  
  
End Class
```

In den Projekteigenschaften sollte Ihnen nun als Startobjekt die Klasse „Sample“ vorgeschlagen werden. In diesem Startobjekt wird für den Anwendungsstart automatisch die „Main“-Methode aufgerufen.

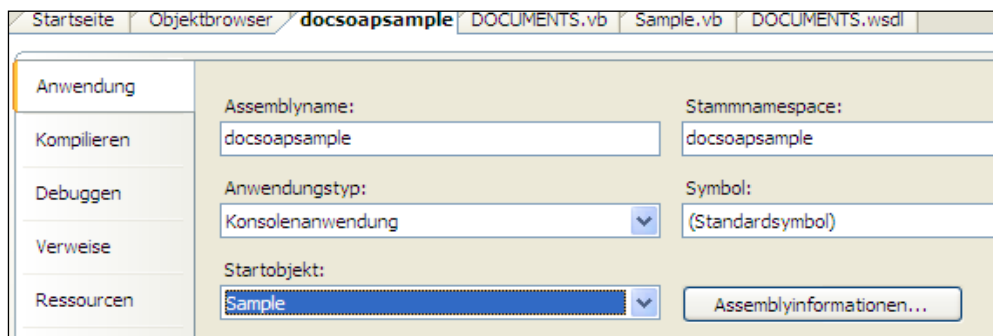


Abb. 5: Auswahl des Startobjektes für das Projekt

In der Main-Methode erstellen Sie ein **DOCUMENTS**-Objekt und spezifizieren die URL, unter der der Proxy Ihre Anfragen entgegennimmt. Anschließend wird eine Anmeldung durchgeführt und die Anzahl der Mappentypen, die für den Benutzer sichtbar sind, ermittelt. Abschließend erfolgt die Abmeldung vom Proxy.

```
Public Shared Sub Main()  
  
    ' Die Schnittstelle instanziiieren  
    Dim doc as New DOCUMENTS  
  
    ' Die URL unter der der Proxy für diese Anwendung erreichbar ist.  
    doc.Url = "http://localhost:11001"  
  
    Try  
        ' Anmeldung beim Proxy  
        ' Sie dürfen nicht mit demselben User bereits am Server angemeldet sein  
        doc.login("schreiber", "peachit", "willi", "", "de")  
    End Try  
  
End Sub
```

```

' Welche Mappentypen gibt es für diesen User auf dem Server?
Dim ftsd() As FileTypeShortDescr

ftsd = doc.getFileTypes(False)

If Not ftsd Is Nothing Then

    Console.WriteLine("Mappenanzahl: {0}", ftsd.length)

End If

Catch ex As Exception

    Console.WriteLine("Exception: {0}", ex.Message)

Finally

    'Abmeldung

    doc.logout

End Try

End Sub

```

Ggfs. muss vor dem Aufruf der „docsoapsample.exe“ die Codepage für die anzeigende Konsole angepasst werden, dies gilt insbesondere wenn der Server UTF-8 unterstützt und der Proxy in seiner Ini-Datei mit der Option "PortalServerEncoding=UTF-8" gestartet wurde.

```
PortalServerEncoding=UTF-8
```

Encoding-Einstellung in der docsoapproxy.ini

Mit dem Befehl "chcp" gefolgt von *einem Code Page Identifier (MSDN)* lässt sich dann die Codepage der Konsole umschalten. Für UTF-8 ergibt sich die Codepage-Nummer 65001.

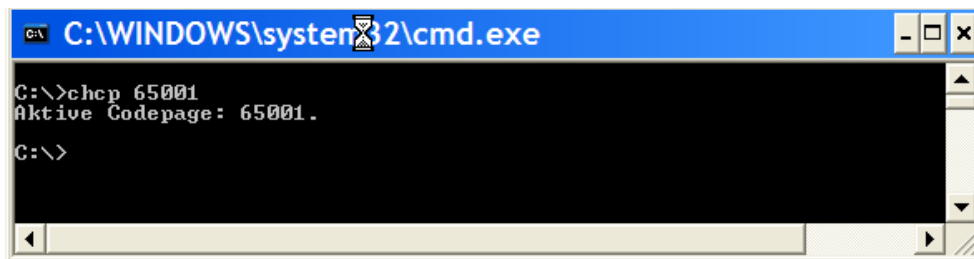


Abb. 6: Einstellen der Unicode - utf-8 Codepage 65001

Damit der Text (z.B. Umlaute) in der aktuellen Codepage auf der Konsole korrekt dargestellt wird muss ggfs. die Konsolenschriftart angepasst werden. Dies kann über "*Titelzeile des Fensters->Kontextmenü->Eigenschaften*" vorgenommen werden.

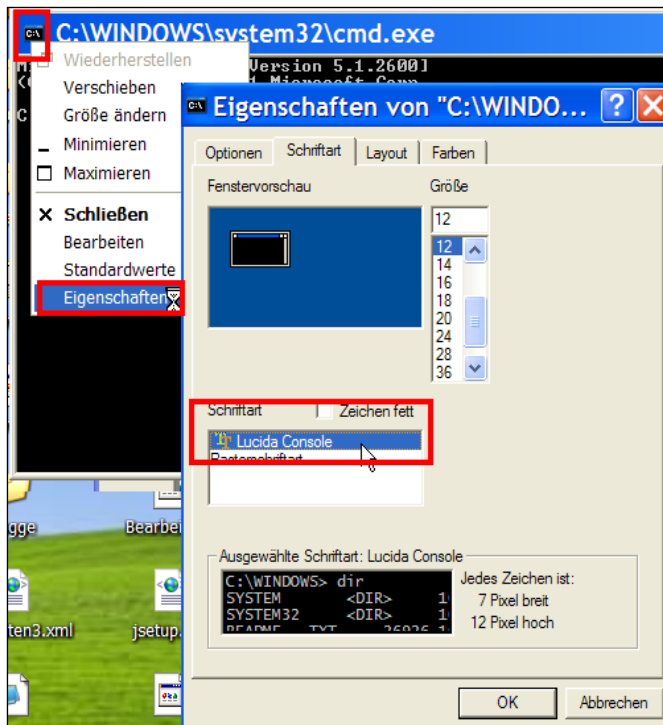


Abb. 7: Einstellen der Konsolenschriftart- Lucida Console

2.7 Erstellung eines gSOAP-Clients mit Visual C++ 2008®

Das gSOAP-Toolkit bietet eine automatisierte SOAP und XML-Datenbindung für C/C++. Es vereinfacht die Entwicklung von SOAP Web Services und Clients. Zur Erstellung eines gSOAP-Clients sind die folgenden Tools benötigt, welche im Verzeichnis `gsoap\bin\win32` im gSOAP aus [SourceForge](#) vorhanden sind:

- Der WSDL-Importer `wsdl2h`.
- Der Stub-Kompiler `soapcpp2`.

Der gSOAP-Importer `wsdl2h` importiert die WSDL eines Web Services und generiert eine Header-Datei, welche die C/C++ Deklarationen der Service-Komponenten enthält. Diese Header-Datei wird dann durch `soapcpp2` kompiliert, um Stubs für den Client zu generieren. Nun können Sie mit der Erstellung eines gSOAP-Clients zu beginnen.

1. Download gSOAP aus [SourceForge](#).
2. Importieren Sie die WSDL durch den Aufruf von `wsdl2h` aus der Kommandozeile:

```
wsdl2h -I C:\gsoap-2.8\gsoap\WS -f -o doc4.h "C:\Program
Files\Documents4\soapproxy\DOCUMENTS.wsdl"
```

Die Option `-I` sagt dem Importer, wo die benötigten gSOAP Include-Dateien zu finden sind. Zur Generierung vom C++ Code wird die Option `-f` benutzt. Die Option `-o` legt die Output-Datei fest. Der letzte Parameter gibt die zu importierende WSDL-Datei der SOAP-API an. Bei einer erfolgreichen Importierung wird die Output-Datei `doc4.h` erzeugt.

3. Generiere C++ Stubs durch den Aufruf von `soapcpp2` aus der Kommandozeile:

```
soapcpp2 -C -L -I C:\gsoap-2.8\gsoap\import -w -x doc4.h
```

Die Option `-C` gibt an, dass nur Code für die Client-Seite generiert wird. `-L` wird benötigt, wenn die Datei `soapClientLib.cpp` nicht generiert werden soll. Die Option `-I` gibt den Pfad der Include-Datei an. `-w` sagt dem Generator, dass die Schema-Dateien nicht erzeugt werden sollen. `-x` teilt ihm mit, die Dateien für die XML-Meldungen nicht zu generieren.

4. Erstellen einer Konsolenanwendung mit Visual C++ 2008®:

Der Einfachheit halber erstellen Sie eine C++ Win32-Konsolenanwendung und akzeptieren dabei alle Voreinstellungen. Die weiteren Schritte sind im Folgenden beschrieben.

- 1) Füge die folgenden von gSOAP generierten Dateien in das Projekt unter Projekt > Vorhandenes Element hinzufügen hinzu:
 - `DOCUMENTS.nsmmap`
 - `soapC.cpp`
 - `soapClient.cpp`
 - `soapDOCUMENTSProxy.h`
 - `soapH.h`
 - `soapStub.h`
- 2) Füge die folgenden Dateien aus dem gSOAP Verzeichnis (`C:\gsoap-2.8\gsoap`) in das Projekt gleichfalls hinzu:
 - `stdsoap2.cpp`
 - `stdsoap2.h`
- 3) Deaktiviere die Verwendung von vorkompilierten Headern für die folgenden Dateien:
 - `soapC.cpp`
 - `soapClient.cpp`
 - `Stdsoap2.cpp`

Aus dem Kontextmenü der Dateien im Projektmappen-Explorer selektiere `Eigenschaften > C/C++ > Vorkompilierte Header`, ändere die Option für `VorKompilierten Header erstellen/verwenden` in `Vorkompilierte Header nicht verwenden`.

- 4) Füge den Include-Pfad zu gSOAP (z.B. `C:\gsoap-2.8\gsoap`) in das Projekt unter Projekt > Eigenschaften > C/C++ > Allgemein > Zusätzliche Includeverzeichnisse hinzu.
- 5) Öffne die C++ Datei der Konsolenanwendung. Füge die folgenden `#includes` in die Datei genau hinter dem `stdafx.h` Include hinzu:
 - `#include "soapDOCUMENTSProxy.h"`
 - `#include "DOCUMENTS.nsmmap"`
 - `#include "stdsoap2.h"`

5. Beispielcode für den gSOAP-Client

```
#include "stdafx.h"

#include "soapDOCUMENTSProxy.h"

#include "DOCUMENTS.nsmap"

#include "stdsoap2.h"

#include<iostream>
#include<fstream>

usingnamespace std;
```

```
int _tmain(int argc, _TCHAR* argv[])
{
    // Create a gSOAP client object.
    DOCUMENTS doc;

    // Endpoint URL of soapproxy (change as needed)
    doc.endpoint = "http://localhost:11001";

    // Sample for login
    _ns2__login req;
    req.locale = "de";
    req.user = "schreiber";
    req.principal = "peachit";
    req.code = "";
    req.passwd = "willi";

    _ns2__loginResponse res;
    int ret = doc.__ns1__login(&req, &res);
    if (ret == SOAP_OK)
        cout << "\nLogin succeeded: " << res.session << endl;
    else
    {
        cerr << "Login failed with the error message: \n"
                << doc.soap->fault->faultstring;
    }

    // Sample for getInbox
    _ns2__getInbox req2;
    _ns2__getInboxResponse res2;
```

```

ret = doc.__ns1__getInbox(&req2, &res2);
if (ret == SOAP_OK)
{
    cout << "\nGetInbox succeeded: " << res2.idFolder << endl;
    ns2__FileStatusList* fileStatuses = res2.fileStatuses;
    unsigned int size = fileStatuses->fileStatus.size();
    if (fileStatuses && size > 0)
    {
        // Sample for getFileInfo: We iterate through
        // all files in the inbox.
        _ns2__getFileInfo req3;
        _ns2__getFileInfoResponse res3;
        req3.allFields = true;
        req3.allAttributes = true;
        for (unsigned int i = 0; i < size; ++i)
        {
            req3.fileId = fileStatuses->fileStatus[i]->fileId;
            ret = doc.__ns1__getFileInfo(&req3, &res3);
            if (ret == SOAP_OK)
            {
                cout << "\n=====\\n";
                cout << "Information for file with id '"
                    << req3.fileId << "':\\n";
                cout << "filetype id: " << res3.filetypeId << endl;
                cout << "filetype name: " << res3.filetypeName << endl;
                cout << "filetype label: " << res3.filetypeLabel << endl;
                if (res3.fieldvalues)
                {
                    cout << "\\nField values: \\n";
                    vector<ns2__FieldData* >* pFieldData =
                        &(res3.fieldvalues->field);

                    vector<ns2__FieldData* >::iterator it;
                    for (it = pFieldData->begin(); it != pFieldData->end(); ++it)
                    {
                        cout << (*it)->name << ": " << (*it)->value << endl;

```

```

    }

}

if (res3.documents)
{
    cout << "\nDoc infos: \n";
    vector<ns2__DocInfo* >* pDocInfo =
        &(res3.documents->document);
    vector<ns2__DocInfo* >::iterator it;
    for (it = pDocInfo->begin(); it != pDocInfo->end(); ++it)
    {
        cout << "Id: " << (*it)->id << ", Name: " << (*it)->name
            << ", Size: " << (*it)->size << endl;

        // Sample for getDocument
        _ns2__getDocument req4; _ns2__getDocumentResponse res4;
        req4.fileId = req3.fileId;
        req4.docId = (*it)->id;
        ret = doc.__ns1__getDocument(&req4, &res4);
        if (ret == SOAP_OK)
        {
            string fileName = "Sample_" + res4.name;
            ofstream outfile(fileName.c_str(), ofstream::binary);

            // Write the document content to file named 'fileName'
            // in the current workspace
            outfile.write((const char*)res4.data->__ptr, res4.size);
            outfile.close();
        }
        else
        {
            cerr << "GetDocument failed with the error message: \n"
                << doc.soap->fault->faultstring;
        }
    }
}
}

```

```

    }

    else
    {
        cerr << "GetFileInfo failed with the error message: \n"
              << doc.soap->fault->faultstring;
    }
}

}

else
{
    cerr << "GetInbox failed with the error message: \n"
          << doc.soap->fault->faultstring;
}

// Sample for logout
_ns2__logout req5;
_ns2__logoutResponse res5;
ret = doc.__ns1__logout(&req5, &res5);
if (ret != SOAP_OK)
{
    cerr << "Logout failed with the error message: \n"
          << doc.soap->fault->faultstring;

    return 0;
}

```

2.8 Erstellung eines WCF-Clients mit Visual C# 2008®

Windows Communication Foundation (WCF) ist ein Tool im .NET Framework (ab Version 3.0) zum Erstellendienst-orientierter Applikationen. Ein WCF-Client kommuniziert mit einem WCF-Dienst über einen Endpunkt des Diensts, welcher sich aus den folgenden Eigenschaften zusammensetzt:

- Eine Adresse als URL mit Angabe, wo der Endpunkt zu finden ist.
- Eine Verbindung, die festlegt, wie die Daten bei der Kommunikation übertragen werden.
- Ein Vertrag, der die verfügbaren Vorgänge definiert.
- Eine Menge von Verhaltensweisen zum Anpassen des lokalen Verhaltens des Dienstendpunts.

Im Folgenden wird schrittweise beschrieben, wie ein WCF-Client mit Visual C# 2008® unter .Net Framework 3.5 erstellt wird.

1. Erstelle eine C# Konsolenanwendung mit dem Projektnamen `docsoapcs`.

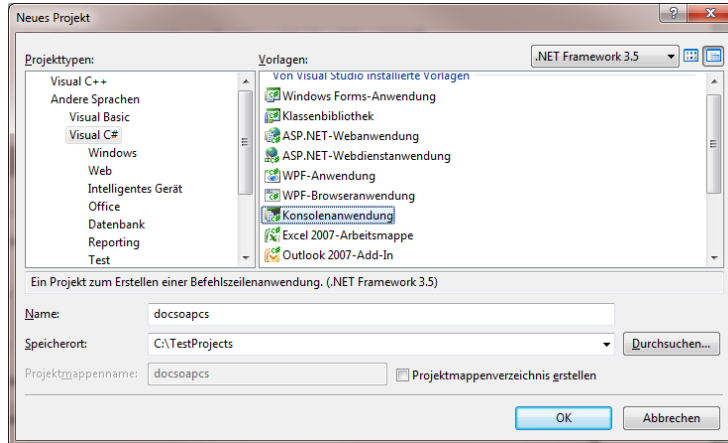


Abb. 8: Erstellen einer C# Konsolenanwendung

2. Öffnen Sie den Dialog `Dienstverweis hinzufügen` unter `Projekt > Dienstverweis hinzufügen`.
3. Im Dialog geben Sie die SOAP API WSDL-Datei (wie z.B. `C:\Program Files\Documents4\soapproxy\DOCUMENTS.wsdl`) in das Eingabefeld `Adresse` ein und spezifizieren Sie einen Namespace (als `ServiceReference1` voreingestellt) für den generierten Code und klicken Sie `OK`.

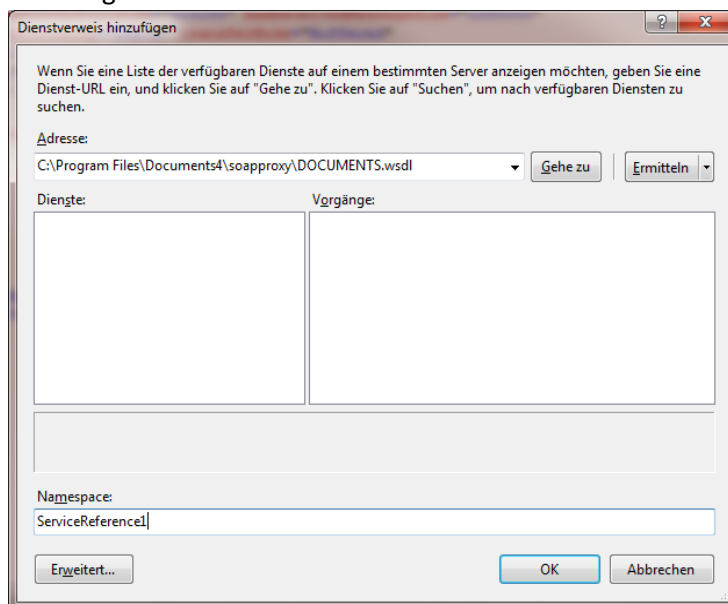


Abb. 9: Hinzufügen eines Dienstverweises

4. Klappen Sie `Service References` im Projektmappen-Explorer auf und doppelklicken Sie auf den oben spezifizierten Namespace (`ServiceReference1`).

Die generierte Clientproxy-Klasse `DOCUMENTSPortTypeClient` unter dem Namespace `docsoapcs.ServiceReferencel` im Objektbrowser zu sehen.

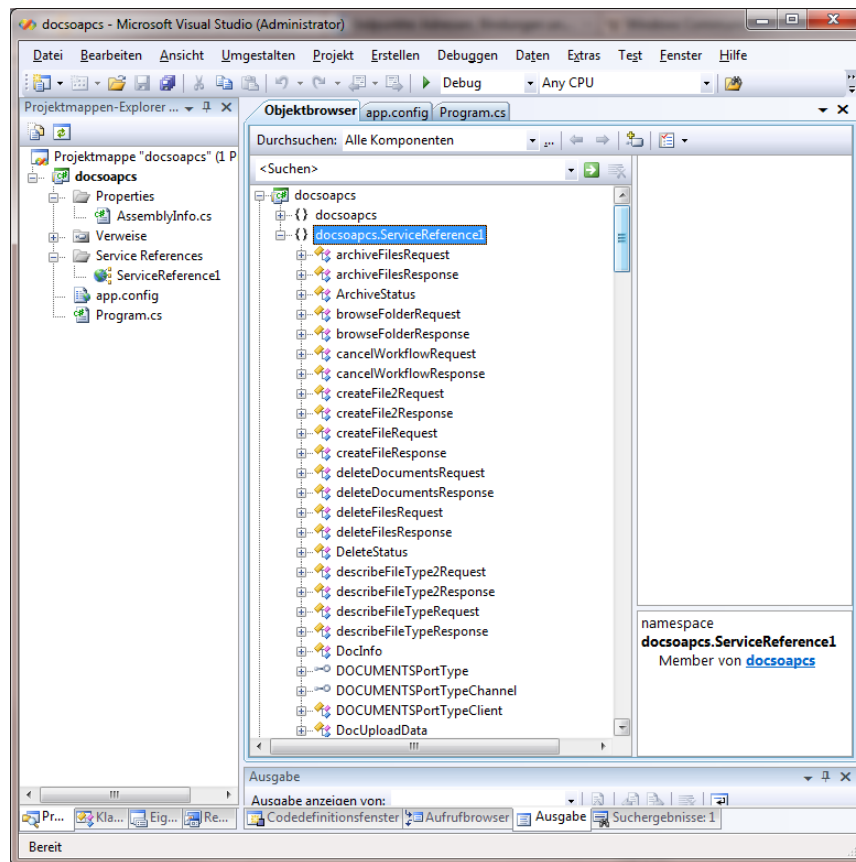


Abb. 10: Veranschaulichung eines Service References

5. Gehen Sie zur C# Datei der Konsolenanwendung (`Program.cs`) und fügen Sie die `using`-Direktive `using docsoapcs.ServiceReferencel` in die Datei ein.
6. Nun können Sie eine Instanz des WCF-Clients erzeugen und dann seine Methoden aufrufen, um auf die Funktionalität des WCF-Dienstszuzugreifen.
7. Beispielcode für den WCF-Client

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.IO;
using docsoapcs.ServiceReferencel;

namespace docsoapcs
{
    Class Program
    {
```

```

Static void Main(string[] args)
{
    // Create a WCF client object with the given client
    // endpoint configuration to invoke the service.
    // Make sure that the URL of the soapproxy is correct.
    // The default is localhost.If the proxy is running
    // on a different machine you have to correct this.

```

```

DOCUMENTSPortTypeClient myDoc = new
DOCUMENTSPortTypeClient("DOCUMENTS", "http://localhost:11001");

string currOperation = null;
string sessionId = null;
try
{
    // Sample for login
    currOperation = "Login";

    myDoc.login("schreiber", "peachit", "willi", "", "de",
    out sessionId);

    Console.WriteLine("\nLogin succeeded: " + sessionId);

    //Sample for getInbox
    currOperation = "GetInbox";

    FileStatus[] fileStatuses;

```

```

string idFolder = myDoc.getInbox(ref sessionId, out fileStatuses);

Console.WriteLine("\nGetInbox succeeded: " + idFolder);

// Sample for getFileInfo: We iterate through
// all files in the inbox
currOperation = "GetFileInfo";
foreach (FileStatus fileStatus in fileStatuses)
{
    DocInfo[] documents;

    string filetypeId;

    string filetypeName;

    string filetypeLabel;

    FieldData[] fieldvalues = myDoc.getFileInfo(ref sessionId,
        fileStatus.fileId, true, true, null, out documents,
        out filetypeId, out filetypeName, out filetypeLabel);

    Console.WriteLine("\n=====");

    Console.WriteLine("Information for file with id '" + fileStatus.fileId
        + "':\n");

```

```

Console.WriteLine("filetype id: " + fileTypeId + "\n");
Console.WriteLine("filetype name: " + fileTypeName + "\n");
Console.WriteLine("filetype label: " + fileTypeLabel + "\n");

Console.WriteLine("\nField values: \n");
foreach (FieldData fieldData in fieldvalues)
{
    Console.WriteLine(fieldData.name + ": " + fieldData.value + "\n");
}

Console.WriteLine("\nDoc infos: \n");
foreach (DocInfo docInfo in documents)
{
    Console.WriteLine("Id: " + docInfo.id + ", Name: " +
        docInfo.name + ", Size: " + docInfo.size + "\n");

    // Sample for getDocument
    currOperation = "GetDocument";

    int size;
    string mime;
    byte[] data;

    string name = myDoc.getDocument(ref sessionId,
        fileStatus.fileId, docInfo.id,
        out size, out mime, out data);

    // Write the document content to file
    // named 'fileName' in the current workspace.
    string fileName = "Sample_" + name;
    FileStream outfile = new FileStream(fileName, FileMode.Create);
    outfile.Write(data, 0, data.Length);
    outfile.Close();
}
}
catch (System.Exception e)
{
    Console.WriteLine("\n" + currOperation + " failed with
        the error message:\n" + e.Message);
}

```

```
finally
{
    // Sample for logout
    if (sessionId != null)
        myDoc.logout(ref sessionId);

    // Close the client
    myDoc.Close();
}
}}}
```

3. API

3.1 DOCUMENTS-Methoden

Documents.archiveFiles

Beschreibung	DOCUMENTS-Mappen archivieren.
Eingabeparameter	fileIds: 0..n String Liste der technischen Bezeichner der zu archivierenden Mappen. deleteOnSuccess: 1 Boolean optional Gibt an, ob erfolgreich archivierte Mappen anschließend in DOCUMENTS gelöscht werden sollen. Wenn der Parameter fehlt, wird dies als false gewertet.
Ausgabeparameter	statusList: 0..n ArchiveStatus Enthält für jede Archivierung Meldungen über den Verlauf.
Beispiel VB	<pre> ' Array with status information about the files which ' are currently in the users inbox Dim fileStatuses() As FileStatus = Nothing Dim idFolder As New String(String.Empty) ' get the inbox files idFolder = doc.getInbox(fileStatuses) ' Now pick the first file If fileStatuses.Length > 0 Then Dim archiveStatus() As ArchiveStatus = Nothing Dim deleteOnSuccess As Boolean = False Dim fileIds() As String = {fileStatuses(0).fileId} ' archive archiveStatus = doc.archiveFiles(fileIds, deleteOnSuccess) ' and check for the status information For Each ars As ArchiveStatus In archiveStatus Console.WriteLine("fileId: {0}", ars.fileId) Console.WriteLine("messages: {0}", ars.messages) Next Else Console.WriteLine("No file to archive") End If </pre>
Erläuterung	Wir holen aus der Inbox die erste Mappe und ermitteln deren Id. Anschließend wird diese Mappe in das Archiv überführt, das ihr über ihren Mappentyp zugeordnet ist. Über den Erfolg des Archivtransfers informiert das zurückgegebene ArchiveStatus-Objekt, dessen Meldungen auf der Konsole ausgegeben werden.
Kommentar	Fehlermeldungen für den Transfer werden im ArchiveStatus aufgeführt.

Documents.browseFolder

Beschreibung	Mappen eines Ordners auflisten.
Eingabeparameter	folderName: 0..1String, optional Der Name eines öffentlichen Ordners. folderId: 0..1 String, optional Der technische Bezeichner eines Ordners kann alternativ zu folderName oder folderType angegeben werden. folderType: 0..1 String, optional Der Typbezeichner eines persönlichen Standardordners kann alternativ zu folderName oder folderId angegeben werden. Eine Liste der erlaubten Bezeichner befindet sich im Anhang Bezeichner für persönliche Standardordner. startIndex: 1 Integer Der Index der ersten auszugebenden Mappe. Die Zählung beginnt dabei mit Nummer 0. Falls startIndex=0 und count=-2, werden alle Mappen des Ordners auf einmal aufgelistet. count: 0..1 Integer, optional Die gewünschte Höchstzahl aufzulistender Mappen muss immer zusammen mit startIndex angegeben werden. Wenn der Wert auf -1 gesetzt wird, verwendet diese Funktion die voreingestellte Mappenanzahl des Benutzers. Wird der Wert auf -2 gesetzt, so werden alle Mappen ab dem startIndex ermittelt. preview: 0..1 Boolean Gibt an, ob auch Feldinhalte zur Vorschau mit ausgegeben werden sollen.
Ausgabeparameter	fieldNames: 1StringList Liste von Feldnamen für die Vorschau. Als Feldnamen für Mappenattribute werden die gleichen Bezeichner verwendet wie im Anhang beschrieben (siehe 5.4 für spezielle Feldbezeichner). Der Parameter entfällt, wenn keine Vorschau (preview) angefordert wurde. Dieser Parameter ist ein reiner Ausgabeparameter, es ist hier nicht möglich, eine Liste von Feldnamen zu übergeben. files: 0..nFolderFile Mappenliste des Ordners bzw. der aktuell abgefragte Ausschnitt davon. previousIndex: 1 Integer Wenn ein Zurückblättern in der Liste möglich ist, dann wird hier der Startindex für die erste Mappe der vorherigen Seite ausgegeben, ansonsten ein negativer Wert. nextIndex: 1 Integer Wenn ein Weiterblättern in der Liste möglich ist, dann wird hier der Startindex für die erste Mappe der nächsten Seite ausgegeben, ansonsten ein negativer Wert.
Rückgabewert	headline: 1 String Die Überschrift für den Ordner.
Beispiel VB	<pre>Dim folderType As New String("Favourites") ' we don't need name and id</pre>

	<pre> Dim folderName As New String(String.Empty) Dim folderId As New String(String.Empty) Dim files() As FolderFile = Nothing Dim headline As String Dim startIndex As Integer = 0 Dim previousIndex As Integer Dim nextIndex As Integer Dim count As Nullable(Of Integer) Dim countSpecified As Boolean = count.HasValue Dim preview As Boolean = False Dim fieldNames As New List(Of String) ' browse headline = doc.browseFolder(folderName, folderId, folderType, _ startIndex, _ count, countSpecified, _ preview, _ fieldNames.ToArray, files, _ previousIndex, nextIndex) Console.WriteLine("headline: {0}", headline) If Not files Is Nothing Then Console.WriteLine("Files found: {0}", files.Length()) ' files in folder For Each ff As FolderFile In files Console.WriteLine("Id:{0}", ff.id) Next End If </pre>
Erläuterung	Wir ermitteln die Mappenliste im Ordner „Favourites“ und identifizieren diese Ordner über den FolderType. Da wir keine Feldliste ausgeben wollen, schalten wir preview auf false. Weiter soll die ganze Liste angefordert werden, wodurch der nullable count-Parameter nicht mit einem Wert zu belegen ist und somit countSpecified automatisch false wird. Nach Aufruf der browse-Funktion iterieren wir durch das zurückgegebene Array von FolderFiles-Objekten und geben die Mappenid auf der Konsole aus.
Kommentar	Die Fehlerstrings bei einer ausgelösten Ausnahme beginnen häufig mit einer technischen Kurzbezeichnung, der durch Pipezeichen abgetrennte Zusatzangaben folgen können. Die Bezeichnungen beginnen immer mit der Zeichenfolge "DlcErr". Manche persönlichen Standardordner können mittlerweile weitere Unterordner enthalten, die diese Funktion bei der Angabe von folderType nicht berücksichtigt. Um auch deren Inhalte erreichbar zu machen, wurde die Funktion Documents.getFolderStructure eingebaut.

Documents.cancelWorkflow

Beschreibung	Beendet den Workflow auf einer Mappe.
Eingabeparameter	fileId : 1 String Id der Mappe, für die ein laufender Workflow gestoppt werden soll.
Ausgabeparameter	Keiner
Rückgabewert	Keiner
Beispiel VB	<pre>Dim fileStatuses() As FileStatus = Nothing doc.getInbox(fileStatuses) If fileStatuses.Length > 0 Then For Each fs As FileStatus In fileStatuses Console.WriteLine("fileId : {0}", fs.fileId) doc.cancelWorkflow(fs.fileId) Next Else Console.WriteLine("No files found") End If</pre>
Erläuterung	Es werden alle Mappen in der Inbox des angemeldeten Benutzers ermittelt und anhand der fileId die cancelWorkflow-Methode auf einem DOCUMENTS-Objekt für jede dieser Mappen aufgerufen.
Kommentar	

Documents.createFile

Beschreibung	Eine neue Mappe erstellen.
Eingabeparameter	filetype: 1 String Der Name des Mappentyps, zu dem eine neue Mappe erzeugt werden soll. Damit die neue Mappe angelegt werden kann, muss der Mappentyp in DOCUMENTS freigegeben sein. Für die Neuanlage einer Archivmappe ist hier der Archivzielbezeichner anzugeben. fields: 0..n FieldData Eine Liste mit FieldData-Objekten der zu schreibenden Mappenfelder, wobei deren Elementnamen den Feldnamen und deren Elementwerte den Feldwerten entsprechen. Alle Werte müssen als String angegeben werden. addDocs: 0..n DocUploadData Eine Liste mit den hinzuzufügenden Dokumenten und den zugehörigen Daten.
Ausgabeparameter	Keiner
Rückgabewert	fileId : 1 String Ein technischer Bezeichner der neuen Mappe. Wenn eine Archivmappe erzeugt wurde, so wird der <i>Archivmappen-Key</i> zurückgegeben.
Beispiel VB	<pre>' the filetype of the new file Dim filetype As New String("ftRecord") ' Building the fields Dim fds As New List(Of FieldData)</pre>

	<pre> Dim remark As New FieldData remark.name = "hrRemarks" remark.value = "The remark" fds.Add(remark) Dim no As New FieldData no.name = "hrEmployeeNo" no.value = "00001" fds.Add(no) ' Create Attachements Dim newFile As New DocUploadData Dim register As New String("Documents") newFile.register = register Dim path As New String("test.txt") newFile.name = path Dim newByteArray() As Byte = System.IO.File.ReadAllBytes(path) newFile.data = newByteArray addDocs.Add(newFile) Dim newFileId As New String(String.Empty) newFileId = doc.createFile filetype, fds.ToArray, addDocs.ToArray) </pre>
Erläuterung	Für den FileType „ftRecord“ wird eine neue Mappe erzeugt. Dazu wird eine Feldliste angelegt, die die einzelnen FieldData-Objekte aufnimmt, deren Feldwerte dann auf der Mappe gesetzt werden sollen. Zusätzlich wird ein Dokument angehängt. Dazu wird die Datei „test.txt“, die in dem Verzeichnis liegen muss, in dem die Anwendung gestartet wurde, als ByteArray eingelesen. Das erzeugte DocUploadData-Objekt wird der Liste hinzugefügt, die dann als Array dem createFile-Aufruf mitgegeben wird.
Kommentar	

Documents.createFile2

Beschreibung	Erstellt ähnlich wie Documents.createFile eine neue Mappe. Allerdings kann diese Funktion auch eine EAS-Archivmappe erstellen ohne eine DOCUMENTS-Mappe zu erzeugen, indem man den Parameter directEAS auf true setzt.
Eingabeparameter	filetype: 1 String Der Name des Mappentyps, zu dem eine neue Mappe erzeugt werden soll. Damit die neue Mappe angelegt werden kann, muss der Mappentyp in DOCUMENTS freigegeben sein. Für die Neuanlage einer Archivmappe ist hier der Archivzielbezeichner anzugeben. directEAS: 1 Boolean Für ein EAS-Archiv kann die Mappenanlage direkt im Archiv erfolgen (true) oder die Archivmappe wird über das DOCUMENTS erzeugt, wobei Defalutwerte gesetzt und ggfs. Scripte ausgeführt werden. fields: 0..n FieldData Eine Liste mit FieldData-Objektender zu schreibenden Mappenfelder, wobei deren Elementnamen den Feldnamen und

	deren Elementwerte den Feldwerten entsprechen. Alle Werte müssen als String angegeben werden. addDocs: 0..n DocUploadData Eine Liste mit den hinzuzufügenden Dokumenten und den zugehörigen Daten.
Ausgabeparameter	Keiner
Rückgabewert	fileId: 1 String Ein technischer Bezeichner der neuen Mappe. Wenn eine Archivmappe erzeugt wurde, so wird der <i>Archivmappen-Key</i> zurückgegeben.
Beispiel VB	<pre>' the filetype of the new file Dim filetype As New String("ftRecord@peachitStore1") ' Building the fields Dim fds As New List(Of FieldData) Dim remark As New FieldData remark.name = "hrRemarks" remark.value = "The remark" fds.Add(remark) Dim directEAS as Boolean directEAS = True Dim newFileId As New String(String.Empty) newFileId = doc.createFile2(filetype, fds.ToArray, addDocs.ToArray)</pre>
Erläuterung	Es wird eine Archivmappe vom Typ „ftRecord“ in einem EAS-Archiv erzeugt. Dazu wird eine Feldliste angelegt, die die einzelnen FieldData-Objekte aufnimmt, deren Feldwerte dann auf der Mappe gesetzt werden sollen. Die Mappe wird direkt im Archiv erzeugt.
Kommentar	Die Verwendung dieser Methode ist performanter als die Verwendung der Funktion Documents.createFile im Fall der direkten Erzeugung einer EAS-Archivmappe.

Documents.createFile3

Beschreibung	Erstellt ähnlich wie Documents.createFile2 eine neue Mappe. Allerdings ermöglicht diese Funktion das Hochladen eines Blobs direkt aus dem Dateisystem. Die Blobs müssen sich in einem Freigabeverzeichnis befinden, das durch die Option BlobBasePath in <i>docsoaproxy.ini</i> festgelegt ist und auf das der SOAPProxy Zugriff haben muss. Anstelle des Base64 kodierten Inhalts wird nur noch der relative Dateipfad übergeben.
Eingabeparameter	filetype: 1 String Der Name des Mappentyps, zu dem eine neue Mappe erzeugt werden soll. Damit die neue Mappe angelegt werden kann, muss der Mappentyp in DOCUMENTS freigegeben sein. Für die Neuanlage einer Archivmappe ist hier der Archivzielbezeichner anzugeben. directEAS: 1 Boolean Für ein EAS-Archiv kann die Mappenanlage direkt im Archiv erfolgen (true) oder die Archivmappe wird über das DOCUMENTS erzeugt, wobei Defalutwerte gesetzt und ggfs. Scripte ausgeführt werden.

	fields: 0..nFieldData Eine Liste mit FieldData-Objektender zu schreibenden Mappenfelder, wobei deren Elementnamen den Feldnamen und deren Elementwerte den Feldwerten entsprechen. Alle Werte müssen als String angegeben werden. addDocs: 0..n DocUploadData3 Eine Liste mit den hinzuzufügenden Dokumenten und den zugehörigen Daten.
Ausgabeparameter	Keiner
Rückgabewert	fileId: 1 String Ein technischer Bezeichner der neuen Mappe. Wenn eine Archivmappe erzeugt wurde, so wird der <i>Archivmappen-Key</i> zurückgegeben.
Beispiel VB	<pre>' the filetype of the new file Dim filetype As New String("ftRecord@peachitStore1") ' Building the fields Dim fds As New List(Of FieldData) Dim remark As New FieldData remark.name = "hrRemarks" remark.value = "The remark" fds.Add(remark) Dim directEAS as Boolean directEAS = False ' Add a document to the register "Documents" Dim addDocs As New List(Of DocUploadData3) Dim sameDoc As New DocUploadData3 sameDoc.register = "Documents" sameDoc.name = "test.txt" sameDoc.path = "test.txt" 'located in BlobBasePath sameDoc.replaceSpecified = False sameDoc.versioningSpecified = False sameDoc.deleteBlobSpecified = True sameDoc.deleteBlob = False addDocs.Add(sameDoc) Dim newFileId As New String(String.Empty) newFileId = doc.createFile3(filetype, fds.ToArray, addDocs.ToArray)</pre>
Erläuterung	Es wird eine mappe vom Typ „ftRecord“ im DOCUMENTS erzeugt. Dazu wird eine Feldliste angelegt, die die einzelnen FieldData-Objekte aufnimmt, deren Feldwerte dann auf der Mappe gesetzt werden sollen. Ebenfalls wird eine Liste von DocUploadData3-Objekten konstruiert, die auf ein Dokument aus dem Freigabeverzeichnis (BlobBasePath) verweist. Diese Listen werden dann als Arrays an die Methode „createFile3“ übergeben.
Kommentar	Die Methode steht erst ab der WSDL DOCUMENTS-4.0.1870 zur Verfügung.

Documents.deleteDocuments

Beschreibung	Dokumente aus einer Mappe löschen.
Eingabeparameter	fileId: 1 String Der technische Bezeichner der Mappe. docIds: 0..n String Die Ids der zu löschenden Dokumente. Diese können über Documents.getFileInfo ermittelt werden.
Ausgabeparameter	Keine
Rückgabewert	status: 0..n DeleteStatus Eine unbestimmte Anzahl von DeleteStatus -Objekten, die über den Erfolg/Misserfolg des Löschvorgangs Auskunft geben.

Beispiel VB	<pre> Dim fileStatuses() As FileStatus = Nothing Dim idFolder As New String(String.Empty) ' Get all files from the inbox idFolder = doc.getInbox(fileStatuses) If fileStatuses.Length <= 0 Then Console.WriteLine("No file in inbox") Return True End If ' and check each file for documents For Each fs As FileStatus In fileStatuses Dim fileId As New String(fs.fileId) Dim allFields As Boolean = False Dim allAttributes As Boolean = False Dim wishedFieldNames As New List(Of String) Dim docinfo() As DocInfo = Nothing Dim fileTypeId As New String(String.Empty) Dim fileTypeNames As New String(String.Empty) Dim fileTypeLabel As New String(String.Empty) Dim fds() As FieldData = Nothing ' getFileInfo for file fds = doc.getFileInfo(fileId, allFields, allAttributes, wishedFieldNames.ToArray, docinfo, fileTypeId, fileTypeNames, fileTypeLabel) If Not docinfo Is Nothing Then ' there could be documents Dim delDocs As New List(Of String) ' get all ids of the attached documents For Each di As DocInfo In docinfo delDocs.Add(di.id) Next If delDocs.Count > 0 Then Dim statuses() As DeleteStatus = Nothing ' delete the documents statuses = doc.deleteDocuments(fileId, delDocs.ToArray) If Not statuses Is Nothing Then For Each st As DeleteStatus In statuses Console.WriteLine("Deleted Document Id: {0} Deleted: {1} Message: {2}", _ st.id, st.deleted, st.messages) Next End If Else Console.WriteLine("No documents found") End If End If Next </pre>
-------------	--

Erläuterung	Wir ermitteln alle Mappen, die sich in der Inbox des angemeldeten Benutzers befinden. Für jede dieser Mappen rufen wir mit der fileId die getFileInfo-Methode auf, um auf dem DocInfo-Array zu prüfen, ob die Mappe über Dokumente verfügt. Die Ids dieser Dokumente sammeln wir dann in einer Löschliste und führen dann die Löschung via deleteDocuments durch. Anschließend wird mit den DeleteStatus-Objekten überprüft, inwieweit diese Löschung erfolgreich war, indem wir den Status auf der Konsole ausgeben.
Kommentar	

Documents.deleteFiles

Beschreibung	Ermöglicht das Löschen von Mappen.
Eingabeparameter	fileIds: 0..n String Die technischen Bezeichner der zu löschenden Mappen. Es können auch <i>Archivmappen-Keys</i> verwendet werden.
Ausgabeparameter	Keine
Rückgabewert	statusList: 0..n DeleteStatus Eine unbestimmte Anzahl von DeleteStatus -Objekten, die über den Erfolg/Misserfolg des Löschvorgangs Auskunft geben.
Beispiel VB	<pre> Dim fileStatuses() As FileStatus = Nothing Dim idFolder As New String(String.Empty) ' get all files from the inbox idFolder = doc.getInbox(fileStatuses) ' Now iterate through that array and the files If fileStatuses.Length > 0 Then For Each fs As FileStatus In fileStatuses Console.WriteLine("fileId : {0}", fs.fileId) Dim fileIds() As String = {fs.fileId} ' delete the file Dim deleteStatus() As DeleteStatus = Nothing deleteStatus = doc.deleteFiles(fileIds) For Each ds As DeleteStatus In deleteStatus Console.WriteLine("Id : {0} deleted: {1} messages: {2}", ds.id, ds.deleted, ds.messages) Next Next Else Console.WriteLine("No files found") End If </pre>
Erläuterung	Wir holen die Mappen des angemeldeten Benutzers, die sich in seinem Eingangsordner befinden. Anschließend iterieren wir über diese Mappen und rufen, nachdem wir die MappeId festgestellt und in das Löscharray eingefügt haben, die DeleteFiles-Methode auf. Der Löschstaus für diesen Vorgang wird dann über die Konsole ausgegeben.
Kommentar	

Documents.deleteFiles2

Beschreibung	Ermöglicht das Löschen von Mappen. Als Erweiterung zu deleteFiles können alle Versionen einer EAS-Archivmappe auf einmal gelöscht werden.
Eingabeparameter	fileids: 0..n String Die technischen Bezeichner der zu löschenden Mappen. Es können auch <i>Archivmappen-Keys</i> verwendet werden. allVersions: Boolean Option zum Löschen aller Versionen der angegebenen Mappen. Dies wird nur von EAS unterstützt und ansonsten ignoriert.
Ausgabeparameter	Keine
Rückgabewert	statusList: 0..n DeleteStatus Eine unbestimmte Anzahl von DeleteStatus -Objekten, die über den Erfolg/Misserfolg des Löschvorgangs Auskunft geben.
Beispiel VC#	<pre> void testDeleteFiles2() { string[] filetypes = { }; string[] archives = { "ftRecord@peachitStore1" }; string[] columns = { "Hit_Id" }; string deleteld; HitList hl = doc.report3(filetypes, archives, columns, " hrRemarks~'testKillAllVersions'", "", ""); if (hl.columns == 1 && hl.rows > 0) { deleteld = hl.hit[0].column[0]; } else { Console.WriteLine("No matching file found for this test."); return; } string[] fileids = new string[1]; fileids[0] = deleteld; DeleteStatus[] dstat = doc.deleteFiles2(fileids, true); Console.Write("id " + dstat[0].id); if (dstat[0].deleted) Console.Write(" deleted."); else { Console.Write(" not deleted. Message: " + dstat[0].messages); } Console.WriteLine(""); } </pre>

Erläuterung	Zuerst suchen wir in dem EAS-Archiv peachitStore1 nach einer Mappe des Typs ftRecord mit dem Schlüsselwort testKillAllVersions im Bemerkungsfeld. Von dem ersten Treffer werden alle Versionen gelöscht. Die Übergabe weiterer Treffer an die Löschfunktion könnte zu Fehlermeldungen im DeleteStatus führen, weil den Treffern nicht anzusehen ist, ob es jeweils weitere Versionen der ersten Mappe oder unabhängige Mappen sind.
Kommentar	Die Methode steht erst ab der WSDL DOCUMENTS-4.0.1880 zur Verfügung.

Documents.describeFileType

Beschreibung	Zeigt Detailinformationen zu einem Mappentypen an. Dies können Archiv-, Feld-, Register- und Workflowinformationen sein.
Eingabeparameter	name: String Der Name des Mappentypen für den die Informationen ermittelt werden sollen. Alternativ kann auch die id verwendet werden. In diesem Falle ist für name dann ein Leerstring zu übergeben. id: String Id des Mappentypes, der untersucht werden soll. Alternativ kann auch der name verwendet werden. In diesem Falle ist für die id dann ein Leerstring zu übergeben. categories: 1 String Der String dient dazu die Daten zu spezifizieren, die angefordert werden sollen. In seiner Abhängigkeit werden bei der FileTypeDescription die korrespondierenden Datenstrukturen aufgefüllt. Dabei bieten sich folgende Möglichkeiten: • „fields“ ○ „enum“ Wenn das Feld vom Type „enum“ ist und dessen Aufzählungswerte mit ausgegeben werden sollen, dann muss „enum“ zusammen mit „fields“ im categories-String angegeben werden – „fields, enum“. Die einzelnen Werte für die Sprachen werden in einem String (1) dargestellt, z.B. „year; de:Jahr; en:Year“. ○ „fieldlabel“ / „fieldlabel.locale“ Mit der Angabe „fields,fieldlabel“ im categories-String werden die vollständigen Labels der Felder ausgegeben, falls „fieldlabel.locale“ nicht angegeben ist. Mit der Angabe „fieldlabel.locale“ werden nun die ergonomischen Namen in der eingestellten Sprache zurückgegeben. Die zurückgegebenen Werte stehen für die beiden Fälle im Member 'fieldlabel' der FieldDescription zur Verfügung, z.B. der Wert „de:Betreff; en:Subject“ für „fieldlabel“ bzw. „Betreff“ für „fieldlabel.locale“ im Fall, dass die Sprache Deutsch selektiert ist. Es ist zu erwähnen, falls die beiden

	(„fieldlabel“ und „fieldlabel.locale“) im categories-String angegeben sind, wird „fieldlabel“ ignoriert. ○ „initialValue“ Soll zusätzlich die Spezifikation zur Vorbelegung eines Feldes mit angezeigt werden, so muss der categories-String neben „fields“ auch den Wert „initialValue“ enthalten“. • „docregisters“ Fügt Informationen zu den Dokumentenregistern in die FileTypeDescription ein. • „archiveinfo“ Fügt Archivinformationen in die FileTypeDescription ein. • „workflowinfo“ Fügt Workflowinformationen in die FileTypeDescription ein. Die einzelnen Möglichkeiten können auch kommasepariert zu einem String zusammengesetzt werden, z.B. „fields,enum, docregisters, archiveinfo“.
Ausgabeparameter	Keine
Rückgabewert	description: 1 FileTypeDescription
Beispiel VB	<pre> Try Dim ftsd() As FileTypeShortDescr ftsd = doc.getFileTypes(False) ' and iterate For Each fsd As FileTypeShortDescr In ftsd If Not fsd Is Nothing Then 'describe this filetype by passing his "name" Dim name As String name = (fsd.name) ' here we want to learn something about the fields in this filetype, their name and the filetype Dim cat As New String("fields") Dim ftd As FileTypeDescription ' the call of the documents method at last ftd = doc.describeFileType(name, "", cat) Dim fields() As FieldDescription fields = ftd.fields Console.WriteLine("Fields count: {0} ", fields.Length()) For Each f As FieldDescription In fields Console.WriteLine("Fields Name: {0} Type: {1} ", f.name, f.type) Next End If 'if not fsd Next ' for each Catch ex As Exception Console.WriteLine("DescribeFileType Message: {0}", ex.Message) Return False End Try </pre>

Erläuterung	Es sollen Informationen zu den Feldern des Mappentyps ausgegeben werden, insbesondere der Name und der Feldtyp. Diese Anforderung wird über den Kategorie-String cat=„fields“ gesteuert, der dafür sorgt, dass in der zurückgegebenen FileTypeDescription das fields-Array aufgefüllt wird. Weiter benötigen wir eine Angabe für welchen Mappentyp dies geschehen soll. Dafür wird der describeFileType-Methode der Name des Mappentyps übergeben. In der FileTypeDescription wird dann durch die Felder iteriert und Name sowie Typ auf der Konsole ausgegeben.
Kommentar	Die Mappentypbeschreibung wird wesentlich durch den „categories“-String gesteuert. Wird dieser nicht richtig belegt, dann werden die resultierenden Datenstrukturen auch nicht mit Werten gefüllt. Die Ausgabe archiveinfo wird für das EAS erst ab der WSDL DOCUMENTS-4.0.1827 unterstützt. Wenn das Label des Mappentyps ermittelt werden soll, verwenden Sie die folgende Funktion: Documents.describeFileType2

Documents.describeFileType2

Beschreibung	Zeigt Detailinformationen zu einem Mappentypen an. Wird wie Documents.describeFileType verwendet, allerdings kann hier auch das Label des Mappentypen in der angemeldeten Sprache ermittelt werden.
Eingabeparameter	name: String Der Name des Mappentypen für den die Informationen ermittelt werden sollen. Alternativ kann auch die id verwendet werden. In diesem Falle ist für name dann ein Leerstring zu übergeben. id: String Id des Mappentypen, der untersucht werden soll. Alternativ kann auch der name verwendet werden. In diesem Falle ist für die id dann ein Leerstring zu übergeben. categories: 1 String Der String dient dazu die Daten zu spezifizieren, die angefordert werden sollen. In seiner Abhängigkeit werden bei der FileTypeDescription die korrespondierenden Datenstrukturen aufgefüllt. Dabei bieten sich folgende Möglichkeiten: • „fields“ ○ „enum“ Wenn das Feld vom Type „enum“ ist und dessen Aufzählungswerte mit ausgegeben werden sollen, dann muss „enum“ zusammen mit „fields“ im categories-String angegeben werden – „fields, enum“. Die einzelnen Werte für die Sprachen werden in einem String (1) dargestellt, z.B. „year; de:Jahr; en:Year“. ○ „fieldlabel“ Mit der Angabe „fields,fieldlabel“ im categories-String werden die vollständigen Labels der Felder ausgegeben, falls „fieldlabel.locale“ nicht angegeben ist. Mit der Angabe „fieldlabel.locale“ werden nun die ergonomischen Namen in der eingestellten

	Sprache zurückgegeben. Die zurückgegebenen Werte stehen für die beiden Fälle im Member „fieldlabel“ der FieldDescription zur Verfügung, z.B. der Wert „de:Betreff; en:Subject“ für „fieldlabel“ bzw. „Betreff“ für „fieldlabel.locale“ im Fall, dass die Sprache Deutsch selektiert ist. Es ist zu erwähnen, falls die beiden („fieldlabel“ und „fieldlabel.locale“) im categories-String angegeben sind, wird „fieldlabel“ ignoriert. ○ „initialValue“ Soll zusätzlich die Spezifikation zur Vorbelegung eines Feldes mit angezeigt werden, so muss der categories-String neben „fields“ auch den Wert „initialValue“ enthalten“. • „docregisters“ Fügt Informationen zu den Dokumentenregistern in die FileTypeDescription ein. • „archiveinfo“ Fügt Archivinformationen in die FileTypeDescription ein. • „workflowinfo“ Fügt Workflowinformationen in die FileTypeDescription ein. Die einzelnen Möglichkeiten können auch kommasepariert zu einem String zusammengesetzt werden, z.B. „fields, enum, docregisters, archiveinfo“.
Ausgabeparameter	Keine
Rückgabewert	description: 1 FileTypeDescription2
Beispiel VB	<pre> Try sampleLogin(doc) ' First we get the filetypes Dim ftsd() As FileTypeShortDescr ftsd = doc.getFileTypes(False) ' and iterate through all returned FileTypeShortDescr For Each fsd As FileTypeShortDescr In ftsd If Not fsd Is Nothing Then Console.WriteLine(New String("-", 10)) Console.WriteLine("FileTypeName: {0}", fsd.name) Dim id As New String(String.Empty) id = fsd.id ' the name is optional if we pass the id Dim name As New String(String.Empty) Dim cat As New String(String.Empty) Dim ftd As FileTypeDescription2 ftd = doc.describeFileType2(name, id, cat) If Not ftd Is Nothing Then Console.WriteLine("Filetype name: {0} label: {1}", ftd.name, ftd.label) End If End If Next </pre>

	<pre> Catch ex As Exception Console.WriteLine("DescribeFileType2: {0}", ex.Message) Return False End Try </pre>
Erläuterung	Es soll das Label eines Mappentyps ausgegeben werden. Diese Anforderung wird über den leeren Kategorie-String cat= „“ gesteuert, der dafür sorgt, dass in der zurückgegebenen FileTypeDescription2 das Label ermittelt wird. Weiter benötigen wir eine Angabe für welchen Mappentyp dies geschehen soll. Dafür wird der describeFileType2-Methode die id des Mappentypes übergeben. In der FileTypeDescription2 werdender Name sowie das Label auf der Konsole ausgegeben.
Kommentar	Die Mappentypbeschreibung wird wesentlich durch den „categories“-String gesteuert. Wird dieser nicht richtig belegt, dann werden die resultierenden Datenstrukturen auch nicht mit Werten gefüllt. Die Ausgabe archiveinfo wird für das EAS erst ab der WSDL DOCUMENTS-4.0.1827 unterstützt. Gleiches gilt für die Methode selbst.

Documents.editFile

Beschreibung	Eine Mappe bearbeiten.
Eingabeparameter	fileId: 0..1 String, optional Der eindeutige Bezeichner zur Identifikation der Mappe muss angegeben werden, wenn nicht stattdessen keyfield, keyvalue und filetype angegeben werden. Für die fileId kann auch ein Archivmappen-Key verwendet werden. filetype: 0..1 String, optional Der Mappentyp oder der Archivzielbezeichner der aktiven bzw. archivierten Mappe. Der Parameter entfällt, wenn fileId angegeben wird, muss aber zusammen mit keyfield und keyvalue verwendet werden. keyfield: 0..1 String, optional Der Name eines Feldes, das als Schlüsselfeld zur Identifikation der Mappe fungiert. Der Parameter entfällt, wenn fileId angegeben wird, muss aber zusammen mit keyvalue und filetype verwendet werden. keyvalue: 0..1 String, optional Der Wert eines Schlüsselfeldes zur Identifikation der Mappe. Der Parameter entfällt, wenn fileId angegeben wird, er muss dann aber zusammen mit keyfield und filetype verwendet werden. fields: 0..n FieldData Die zu überschreibenden Mappenfelder als Anzahl von Fielddata-Objekten, wobei deren Elementnamen den Feldnamen und deren Elementwerte den Feldwerten entsprechen. Alle Werte müssen als String angegeben werden. addDocs: 0..n DocUploadData Eine Liste mit den hinzuzufügenden Dokumenten und den dazu notwendigen Daten.
Ausgabeparameter	Keine

Rückgabewert	newId Es wird die fileId der editierten Mappe zurückgegeben. Diese kann sich bei Archivmappen ändern, sofern eine Versionierung der Mappe erfolgte.
Beispiel VB	<pre>Dim filetype As New String(String.Empty) Dim keyField As New String(String.Empty) Dim keyValue As New String(String.Empty) ' we add no documents Dim addDocs As New List(Of DocUploadData) ' the fielddata list with the fielddata to change Dim fds As New List(Of FieldData) Dim rm As New FieldData rm.name = "hrRemarks" rm.value = "The new remark" fds.Add(subject) doc.editFile(fileId, filetype, keyField, keyValue, fds.ToArray, addDocs.ToArray)</pre>
Erläuterung	Bekannt ist die fileId einer Mappe, die zum Mappentyp „crmNote“ gehört. Wir bauen eine Liste von FieldData-Objekten auf, die den Namen „name“ des Feldes „hrRemarks“ und den neuen Wert „value“ für das Feld aufnimmt „The new remark“. Diese Liste wird dann als Array an die editFile-Methode übergeben.
Kommentar	Achten Sie darauf, dass die Identifizierung einer Mappe entweder über die „fileId“ oder über die Kombination „filetype + keyfield + keyvalue“ in eindeutiger Weise erfolgt. Allerdings kann eine Archivmappe bis einschließlich WSDL DOCUMENTS-4.0.1827 nur über einen Archivmappen-Key identifiziert werden, d.h. der Parameter „fileId“ muss angegeben werden. Falls die Identifizierung einer Mappe über die „fileId“ erfolgt und keine Mappe gefunden wird, wird ein Fehler geworfen. Wenn keine Mappe über die Kombination „filetype + keyfield + keyvalue“ gefunden wird, so wird automatisch eine neue aktive Mappe bzw. eine neue Archivmappe (bis einschließlich WSDL DOCUMENTS-4.0.1827) mit den Default-Werten erstellt. Bei Archivmappen wird ab WSDL DOCUMENTS-4.0.1870 allerdings in diesem Fall ein Fehler geworfen. Das Überschreiben der Feldwerte geschieht auch, wenn neuer und alter Wert gleich sind. Beachten Sie bei Verwendung eines Archivmappen-Key das jeweilig unterschiedliche Format bei EE.i, EE.x und EAS Archiven. Um ein Dokument zu versionieren bzw. zu ersetzen, ist die Methode Documents.editFile2 vorgesehen.

Documents.editFile2

Beschreibung	Eine Mappe ähnlich wie Documents.editFile bearbeiten. Allerdings kann man mit dieser Methode ein Dokument in einer aktiven Mappe durch ein gleichnamiges Dokument versionieren bzw. ersetzen. Für Archivmappen funktioniert diese Methode identisch wie Documents.editFile.
Eingabeparameter	fileId: 0..1 String, optional Der eindeutige Bezeichner zur Identifikation einer Mappe muss angegeben werden, wenn nicht stattdessen keyfield, keyvalue und filetype angegeben werden. Für die fileId kann auch ein Archivmappen-Key verwendet werden. filetype: 0..1 String, optional Der Mappentyp oder der Archivzielbezeichner der aktiven bzw. archivierten Mappe. Der Parameter entfällt, wenn fileId angegeben wird, muss aber zusammen mit keyfield und keyvalue verwendet werden. keyfield: 0..1 String, optional Der Name eines Feldes, das als Schlüsselfeld zur Identifikation der Mappe fungiert. Der Parameter entfällt, wenn fileId angegeben wird, muss aber zusammen mit keyvalue und filetype verwendet werden. keyvalue: 0..1 String, optional Der Wert eines Schlüsselfeldes zur Identifikation der Mappe. Der Parameter entfällt, wenn fileId angegeben wird, er muss dann aber zusammen mit keyfield und filetype verwendet werden. fields: 0..n FieldData Die zu überschreibenden Mappenfelder als Anzahl von FieldData-Objekten, wobei deren Elementnamen den Feldnamen und deren Elementwerte den Feldwerten entsprechen. Alle Werte müssen als String angegeben werden. addDocs: 0..n DocUploadData2 Eine Liste mit den hinzuzufügenden Dokumenten und den dazu notwendigen Daten.
Ausgabeparameter	Keine
Rückgabewert	newId Es wird die fileId der editierten Mappe zurückgegeben. Diese kann sich bei Archivmappen ändern, sofern eine Versionierung der Mappe erfolgte.
Beispiel VB	<pre>Dim filetype As New String(String.Empty) Dim keyField As New String(String.Empty) Dim keyValue As New String(String.Empty) ' the fielddata list with the fielddata to change Dim fds As New List(Of FieldData) Dim rm As New FieldData rm.name = "hrRemarks" rm.value = "The new remark" fds.Add(rm) ' Add a document with the same name as the document "test.txt" ' already existing in the register "Documents" Dim addDocs As New List(Of DocUploadData2)</pre>

	<pre>Dim path As New String("test.txt") Dim newByteArray() As Byte = System.IO.File.ReadAllBytes(path) Dim sameDoc As New DocUploadData2 sameDoc.register = "Documents" sameDoc.name = "test.txt" sameDoc.data = newByteArray sameDoc.replaceSpecified = True sameDoc.replace = True sameDoc.versioningSpecified = True sameDoc.versioning = True addDocs.Add(sameDoc) doc.editFile2(fileId, filetype, keyField, keyvalue, fds.ToArray, addDocs.ToArray)</pre>
Erläuterung	Bekannt ist die fileId einer Mappe, die zum Mappentyp „ftRecord“ gehört. Wir bauen eine Liste von FieldData-Objekten auf, die ein Feld mit dem Namen „hrRemarks“ und dem neuen Wert „The new remark“ aufnimmt. Ebenfalls wird eine Liste von DocUploadData2-Objekten konstruiert, die ein Dokument mit demselben Namen wie ein vorhandenes Dokument in der Mappe enthält. Diese Listen werden dann als Arrays an die editFile2-Methode übergeben.
Kommentar	Mit dieser Methode kann ein Dokument in einer aktiven Mappe durch ein gleichnamiges Dokument versioniert bzw. ersetzt werden. Achten Sie darauf, dass die Identifizierung einer aktiven Mappe entweder über die „fileId“ oder über „filetype + keyfield + keyvalue“ in eindeutiger Weise erfolgt. Beispiele der Angabe des Parameters „filetype“ für eine Archivmappe (siehe Archivzielbezeichner): EAS: Filetype7@myStore EE.i: \$(#STANDARD)\REGTEST@eei1 EE.x: Unit=Default/Instance=Default/View=REGTEST*Unit=Default/Instance=Default/DocumentSchema=REGTEST@eex1 Falls die Identifizierung einer Mappe über die „fileId“ erfolgt und keine Mappe gefunden wird, wird ein Fehler geworfen. Wenn keine aktive Mappe über die Kombination „filetype + keyfield + keyvalue“ gefunden wird, so wird automatisch eine neue Mappe mit den Default-Werten erstellt. Bei Archivmappen wird allerdings in diesem Fall ein Fehler geworfen. Das Überschreiben der Feldwerte geschieht auch, wenn neuer und alter Wert gleich sind. Beachten Sie bei Verwendung eines Archivmappen-Key das jeweilig unterschiedliche Format bei EE.i, EE.x und EAS Archiven. Die Methode steht erst ab der WSDL DOCUMENTS-4.0.1870 zur Verfügung.

Documents.editFile3

Beschreibung	Eine Mappe ähnlich wie mit Documents.editFile2bearbeiten. Allerdings ermöglicht diese Funktion das Hochladen eines Blobs direkt aus dem Dateisystem. Die Blobs müssen sich in einem Freigabeverzeichnis befinden, das durch die Option BlobBasePath in docsoaproxy.ini festgelegt ist, und auf das der SOAPProxy Zugriff hat. Anstelle des Base64 kodierten Inhalts wird nur noch der relative Dateipfad übergeben.
Eingabeparameter	fileId: 0..1 String, optional Der technische Bezeichner der Mappe muss angegeben werden, wenn nicht stattdessen keyfield, keyvalue und filetype angegeben werden. Als fileId kann auch ein Archivmappen-Key verwendet werden. filetype: 0..1 String, optional Der Mappentyp oder der Archivzielbezeichner der aktiven bzw. archivierten Mappe. Der Parameter entfällt, wenn fileId angegeben wird, muss aber zusammen mit keyfield und keyvalue verwendet werden. keyfield: 0..1 String, optional Der Name eines Feldes, das als Schlüsselfeld zur Identifikation der Mappe fungiert. Der Parameter entfällt, wenn fileId angegeben wird, muss aber zusammen mit keyvalue und filetype verwendet werden. keyvalue: 0..1 String, optional Der Wert eines Schlüsselfeldes zur Identifikation der Mappe. Der Parameter entfällt, wenn fileId angegeben wird, er muss dann aber zusammen mit keyfield und filetype verwendet werden. fields: 0..n FieldData Die zu überschreibenden Mappenfelder als Anzahl von Fielddata-Objekten, wobei deren Elementnamen den Feldnamen und deren Elementwerte den Feldwerten entsprechen. Alle Werte müssen als String angegeben werden. addDocs: 0..n DocUploadData3 Eine Liste mit den hinzuzufügenden Dokumenten und den dazu notwendigen Daten.
Ausgabeparameter	Keine
Rückgabewert	newId Es wird die fileId der editierten Mappe zurückgegeben. Diese kann sich bei Archivmappen ändern, sofern eine Versionierung der Mappe erfolgte.
Beispiel VB	<pre>Dim filetype As New String(String.Empty) Dim keyField As New String(String.Empty) Dim keyValue As New String(String.Empty) ' the fielddata list with the fielddata to change Dim fds As New List(Of FieldData) Dim rm As New FieldData rm.name = "hrRemarks" rm.value = "The new remark" fds.Add(rm)</pre>

	' Add a document with the same name as the document "test.txt" ' already existing in the register "Documents" Dim addDocs As New List(Of DocUploadData3) Dim sameDoc As New DocUploadData3 sameDoc.register = "Documents" sameDoc.name = "test.txt" sameDoc.path = "test_new.txt" 'located in BlobBasePath sameDoc.replaceSpecified = True sameDoc.replace = True sameDoc.versioningSpecified = True sameDoc.versioning = True sameDoc.deleteBlobSpecified = True sameDoc.deleteBlob = False addDocs.Add(sameDoc) doc.editFile3(fileId, filetype, keyField, keyValue, fds.ToArray, addDocs.ToArray)
Erläuterung	Bekannt ist die fileId einer Mappe, die zum Mappentyp „ftRecord“ gehört. Wir bauen eine Liste von FieldData-Objekten auf, die ein Feld mit dem Namen „hrRemarks“ und dem neuen Wert „The new remark“ aufnimmt. Ebenfalls wird eine Liste von DocUploadData3-Objekten konstruiert, die ein Dokument mit demselben Namen wie ein vorhandenes Dokument in der Mappe enthält. Diese Listen werden dann als Arrays an die editFile3-Methode übergeben.
Kommentar	Siehe den Kommentar aus Documents.editFile2. Die Methode steht erst ab der WSDL DOCUMENTS-4.0.1870 zur Verfügung.

Documents.followUp

Beschreibung	Eine Mappe aus dem Eingangskorb des Benutzers in den Wiedervorlageordner verschieben.
Eingabeparameter	fileId: 1 String Die Id der Mappe. timeAbsolute: 0..1 DateTime.iso8601 Der Zeitpunkt, an dem die Mappe wieder im Eingangskorb erscheinen soll. Wenn der Zeitpunkt in der Vergangenheit liegt, bricht die Funktion mit einem Fehler ab. Es sollen nur Zeiten mit 15,30,45 und 00 Minuten verwendet werden. timeAbsoluteSpecified: 1 Boolean Teilt dem SOAP-Client lediglich mit, dass timeAbsolute mit einem Wert belegt wurde. isUTC: 1 Boolean Angabe, ob die Zeitangabe sich auf die internationale Normalzeit (UTC) bezieht, und folglich erst in die lokale Zeitzone des Portalservers umgerechnet werden muss. comment: 1String Kommentar oder Begründung für das Zurücklegen der Mappe.
Ausgabeparameter	Keiner
Rückgabewert	Keiner

Beispiel VB	<pre>Dim newTime As New System.DateTime newTime = System.DateTime.Now newTime = newTime.AddMinutes(50) Dim followUpTime As Nullable(Of System.DateTime) followUpTime = newTime ' the followUpTime is in local time Dim isUTC As Boolean = False Dim comment As New String("New FollowUp") ' move the file in the follow up folder doc.followUp(fileId, followUpTime, followUpTime.HasValue, isUTC, comment)</pre>
Erläuterung	Wir erzeugen ein DateTime-Objekt mit der aktuellen Zeit und addieren 50 Minuten hinzu. Die Wiedervorlagezeit wird dann mit dieser Zeit belegt, womit der Zeitwert in lokaler Zeit vorliegt, da isUTC mit false belegt wurde. Da followUpTime seinem Typ nach nullable ist, benötigt der Aufruf in VB noch den Parameter followUpTime.HasValue, der der Schnittstelle mitteilt, ob tatsächlich ein Wert vergeben wurde oder nicht. Nach Vergabe eines Kommentars, wird die followUp-Methode auf einem DOCUMENTS-Objekt aufgerufen und die Datei in den Wiedervorlageordner verschoben.
Kommentar	• Je nachdem wie aus der wsdl die Klassen generiert werden, kann die Signatur der Methode abweichen. • Besitzt die Mappe bereits eine Wiedervorlagezeit und es soll eine neue Zeit gesetzt werden, so muss der Zeitabstand mindestens 15 Minuten betragen. • Es sind nur absolute Zeitangaben möglich. Relative Angaben wie "+1 Stunde" hingegen nicht.

Documents.forceLogout

Beschreibung	Führt eine Zwangsabmeldung für die angegebene Session aus.
Eingabeparameter	sessionId : 1 String Die ID der abzumeldenden Session.
Ausgabeparameter	Keiner
Rückgabewert	Keiner
Beispiel VB	<pre>Dim doc as New DOCUMENTS Dim session As New String(String.Empty) session = doc.login("schreiber","peachit","willi","", "de") 'do something ... 'Client-timeout occurred, the session is lost. 'Using forceLogout() to log out the session Dim doc2 as New DOCUMENTS doc2.forceLogout(session)</pre>
Erläuterung	Durch ein Client-Timeout (siehe 5.8) wird eine Session verloren. Bevor das Session-Timeout des SOAPProxy (siehe Abb. 3) greift, kann man mit dieser Methode eine Zwangsabmeldung durchführen, falls diese Session keine Serveraktion mehr durchführt.

	Um diese Methode aufzurufen, muss man nicht angemeldet sein.
Kommentar	Die Methode steht erst ab der WSDL DOCUMENTS-4.0.1880 zur Verfügung.

Documents.getAutoText

Beschreibung	Ermittelt Autotexte (z.B. %createdAt%) für eine bestimmte Mappe.
Eingabeparameter	fileId: 1 String Die Mappe zu der Autotexte ermittelt werden. Es kann auch ein Leerstring verwendet werden. autoTextNames: 0..n String
Ausgabeparameter	Keine
Rückgabewert	autoTextValues: 0..n String Stringarray mit den ermittelten Autotexten.
Beispiel VB	<pre>'we have access to a fileId Dim autoTextNames As New List(Of String) Dim autoTextValues As String() autoTextNames.Add("%fileOwner%") autoTextNames.Add("%title%") autoTextNames.Add("%fileType%") autoTextValues = doc.getAutoText(fileId, autoTextNames.ToArray()) Dim i As Integer = 0 For Each s As String In autoTextValues Console.WriteLine("{0} - {1}", autoTextNames.Item(i), s) i = i + 1 Next</pre>
Erläuterung	Unter Verwendung einer fileId werden die Autotexte zum Mappentitel, Mappentyp und Eigentümer der Mappe ausgelesen und aufsgegen.
Kommentar	Eine Übersicht der Autotexte findet sich in der separaten Dokumentation "DOCUMENTS AutoTexte". Dabei können nicht alle Autotexte im Soapzugriff verwendet werden, dies gilt insbesondere für den Abschnitt 3 "AutoTexte für Aufzählungsfelder" Wenn AutoTexte aus globalen Informationen und System-Variablen ermittelt werden sollen, wie z.B. %currentDate%, %currentWeekday% oder %runscript:Scriptname% , so kann ein Leerstring "" als fileId verwendet werden, da die ermittelten Werte nicht von den Werten einer konkreten Mappe abhängen.

Documents.getDocument

Beschreibung	Daten und Dokumentinformationen zu einem Dokument holen.
Eingabeparameter	fileId: 1 String Der eindeutige technische Bezeichner der Mappe.Es kann auch ein <i>Archivmappen-Key</i> verwendet werden. Die

	docId Dokumente kann über Documents.getFileInfo ermittelt werden. docId: String Der eindeutige technische Bezeichner des Dokuments.
Ausgabeparameter	docId: 1 String Sofern kein Fehler aufgetreten ist, wird der gleichnamige Parameter vom Aufruf wieder zurückgesendet. size: 1 Integer Die Dateigröße in Bytes. mime: 1 String Der Mimetype der Datei, falls bekannt. data: 1 Base64 in VB Array of Byte Der Dokumentinhalt.
Rückgabewert	name: 1String Der Dateiname inklusive Erweiterung.
Beispiel VB	<pre>Dim theFileName As New String(String.Empty) Dim size As New Integer Dim mime As New String(String.Empty) Dim data() As Byte = Nothing ' get the document theFileName = doc.getDocument(fileId, docId, size, mime, data) Console.WriteLine("Name: {0} size: {1} mime: {2}", theName, size, mime) Dim path As New String(theFileName) fs = System.IO.File.Create(path) fs.Write(data, 0, data.Length) fs.Close()</pre>
Erläuterung	Vorausgesetzt wird, dass die fileId der Mappe und die docId des zu extrahierenden Dokumentes bekannt sind. Mit Aufruf der getDocument-Methode wird das Byte-Array mit den Dokumentdaten gefüllt. Über File.Create wird dann auf dem Dateisystem eine neue Datei angelegt und die Bytes aus dem data-Array in die Datei geschrieben.
Kommentar	

Documents.getFileId

Beschreibung	Ermittel die Mappen-Id eines temporären Suchtreffers für einen Archiv-Mappe.
Eingabeparameter	Key 1 String Archivmappenkey einer archivierten Mappe
Ausgabeparameter	Keine
Rückgabewert	fileId: 1 String Die Mappen-Id des korrespondierenden temporären Suchtreffers
Beispiel VB	<pre>Try Dim archives As New List(Of String) archives.Add("ftRecord@peachitStore1") ' additional filetypes to be searched Dim filetypes As New List(Of String)</pre>

	<pre> filetypes.Add("ftRecord") Dim columns As New List(Of String) columns.Add("Hit_ArchiveKey") ' the filter for a fulltext search Dim filter As New String(String.Empty) ' all files filter = "Search_Fulltext~'fa'" Dim sort As New String(String.Empty) ' Search archive Dim hitl As New HitList hitl = doc.report3(filetypes.ToArray, archives.ToArray, columns.ToArray, filter, sort, "Standard") Dim rows As Integer = hitl.rows Dim cols As Integer = hitl.columns If Not hitl Is Nothing Then Dim hds() As HitData = hitl.hit For i As Integer = 0 To rows - 1 Dim hd As HitData = hds(i) For j As Integer = 0 To cols - 1 If Not String.IsNullOrEmpty(hd.column(j)) Then Console.WriteLine(New String("-", 10)) Console.WriteLine("Archived file {0} ", hd.column(j)) Dim fileId As String fileId = doc.getFileId(hd.column(j)) Console.WriteLine("File id {0} ", fileId) End If Next Next i End If Catch ex As Exception Console.WriteLine("Message: {0}", ex.Message) End Try </pre>
Erläuterung	Es wird eine EAS-Archivsuche durchgeführt und der Archivmappenkey der Suchtreffer ermittelt. Für diese wird dann die Maapen-Id der temporären Suchtreffer in Documents ermittelt.
Kommentar	Die Methode steht erst ab der WSDL DOCUMENTS-4.0.1827 zur Verfügung.

Documents.getFileInfo

Beschreibung	Mappendaten zu Feldwerten, angehängten Dokumenten und zugehörigen Mappentyp ermitteln.
Eingabeparameter	fileId: 1 String Der eindeutige Bezeichner der zu untersuchenden Mappe. Es kann auch ein <i>Archivmappen-Key</i> verwendet werden. allFields: 1 Boolean Zeigt an, ob alle Felder geholt werden sollen. allAttributes: 1 Boolean

	Wenn true, dann werden auch erweiterte Felder wie „DlcFile_Owner“ geholt. Siehe Anhang 5.4 für spezielle Feldbezeichner. wishedFieldNames: 0..n String Die Namen der auszugebenden Felder. Die Liste wird ignoriert, wenn allFields den Wert „true“ besitzt.
Ausgabeparameter	filetypeid: 1 String Die Id des Mappentypen, zu dem die Mappe gehört. fileName: 1 String Der Name des Mappentypen für diese Mappe. fileTypeLabel: 1 String Der ergonomische Namen des Mappentypes auf der Weboberfläche. documents: 0..n DocInfo Die Dokumente der Mappe mit Informationen zu Größe, Name und Id.
Rückgabewert	fieldvalues: 0..n FieldData Enthält die Name/Wert Paare zu den einzelnen Mappenfeldern.
Beispiel VB	<pre> Dim fileStatuses() As FileStatus = Nothing doc.getInbox(fileStatuses) If fileStatuses.Length <= 0 Then Console.WriteLine("No files found") Return True End If Dim fileId As New String(fileStatuses(0).fileId) Dim allFields As Boolean = True Dim allAttributes As Boolean = True Dim wishedFieldNames As New List(Of String) Dim fileTypeId As New String(String.Empty) Dim fileName As New String(String.Empty) Dim fileTypeLabel As New String(String.Empty) Dim fds() As FieldData = Nothing Dim docinfo() As DocInfo = Nothing ' getFileInfo fds = doc.getFileInfo(fileId, allFields, allAttributes, wishedFieldNames.ToArray, docinfo, fileTypeId, fileName, fileTypeLabel) Console.WriteLine("FileTypeId: {0} FileName: {1} FileTypeLabel", fileTypeId, fileName, fileTypeLabel) If Not fds Is Nothing Then For Each fd As FieldData In fds ' print fielddata Console.WriteLine("Field: {0} Value: {1}", fd.name, fd.value) Next End If ' fds If Not docinfo Is Nothing Then ' print docinfo For Each di As DocInfo In docinfo </pre>

	<pre> Console.WriteLine("DocName: {0} DocSize: {1} DocId: {2}", di.name, di.size, di.id) Next Else Console.WriteLine("No docinfo") End If 'docinfo </pre>
Erläuterung	Wir holen uns eine Mappe aus der Inbox des angemeldeten Benutzers. Zu dieser Mappe holen wir uns die Informationen zu allen verfügbaren Feldern, den erweiterten Feldern, den Mappentypen, sowie der angehängten Dokumente. Die jeweiligen Werte werden über die Konsole ausgegeben.
Kommentar	Falls keine Feldwerte bzw. Dokumente ausgegeben werden, entfallen die Ausgabeparameter fieldvalues bzw. documents komplett.

Documents.GetFilesInfo

Beschreibung	Mappendaten zu Feldwerten, angehängten Dokumenten und zugehörigen Mappentyp ermitteln.
Eingabeparameter	fileId: 1 String Der eindeutige Bezeichner der zu untersuchenden Mappe. allFields: 1 Boolean Zeigt an, ob alle Felder geholt werden sollen. allAttributes: 1 Boolean Wenn true, dann werden auch erweiterte Felder wie „DlcFile_Owner“ geholt. Siehe Anhang 5.4 für spezielle Feldbezeichner. wishedFieldNames: 0..n String Die Namen der auszugebenden Felder. Die Liste wird ignoriert, wenn allFields den Wert „true“ besitzt. failOnAllErrors 1 Boolean Wenn true wird mit Auftreten eines Fehlers eine Exception geworfen und die Bearbeitung beendet.
Ausgabeparameter	Keine
Rückgabewert	fileinfos: 0..n FileInfo Ermittelt die zu den Mappen gehörenden FileInfo-Objekte, die die Detailinformationen zu den Mappen kapseln.
Beispiel VB	<pre> Dim fileStatuses() As FileStatus = Nothing Dim fids As New List(Of String) idFolder = doc.getInbox(fileStatuses) ' Now iterate through that array and collect the filelds For Each fs As FileStatus In fileStatuses fids.add(fs.fileld) Next Dim allFields As Boolean = True Dim allAttributes As Boolean = True Dim wishedFieldNames As New List(Of String) Dim documents() As DocInfo = Nothing Dim fileTypeNames As New String(String.Empty) Dim failOnAllErrors As Boolean = False Dim fis() As FileInfo = Nothing </pre>

	<pre> ' getFileInfo fis = doc.getFileInfo(fids.ToArray, allFields, allAttributes, wishedFieldNames.ToArray, failOnAllErrors) For Each fi As FileInfo In fis Console.WriteLine("FileName: {0}, fi.fileName) For Each fd As FieldData In fi.fieldvalues Console.WriteLine("Field: {0} Value: {1}", fd.name, fd.value) Next For Each docinfo As DocInfo In fi.documents Console.WriteLine("DocName: {0} DocSize: {1}, docinfo.name, docinfo.size, docinfo.id) Next Next </pre>
Erläuterung	Wir holen uns die Mappen aus der Inbox des angemeldeten Benutzers und sammeln deren Ids. Zu diesen Mappen holen wir uns die Informationen für alle Felder sowie der angehängten Dokumente. Dabei soll auch bei eventuell auftretenden Fehlern, wie z.B. "Mappe nicht gefunden", die Werte für die anderen Mappen ermittelt werden. Die jeweiligen Werte werden über die Konsole ausgegeben.
Kommentar	Siehe auch Documents.getFileInfo .

Documents.getFileTypes

Beschreibung	Ermittelt die Mappentypen für den angemeldeten Benutzer.
Eingabeparameter	ignoreRights : 1 Boolean Gibt an, ob auch die Namen jener Mappentypen ausgegeben werden sollen, für die das verwendete Benutzerkonto keine Leseberechtigung hat.
Ausgabeparameter	Keine
Rückgabewert	0..n FileTypeShortDescr Eine unbestimmte Anzahl an FileTypeShortDescr -Objekten, die den Namen und die Id des Mappentypen enthalten.
Beispiel VB	<pre> Dim doc as New DOCUMENTS Try Dim ftsd() As FileTypeShortDescr ftsd = doc.getFileTypes(ignoreRights) For Each sd As FileTypeShortDescr In ftsd Console.WriteLine("FileType Name: {0} Id: {1}", sd.name, sd.id) Next Catch ex As Exception Console.WriteLine("Logout Message: {0}", ex.Message) End Try </pre>
Erläuterung	Auf einem DOCUMENTS-Objekt werden die FileTypes unter Missachtung spezieller Benutzerrechte ermittelt und in einem Array von FileTypeShortDescr-Objekten abgelegt. Anschließend wird durch das Array iteriert und Id sowie Name des ermittelten FileTypes auf der Konsole ausgegeben.
Kommentar	

Documents.getFilingPlans

Beschreibung	Ermittelt die Aktenpläne für den angemeldeten Benutzer.
Eingabeparameter	Keiner
Ausgabeparameter	Keine
Rückgabewert	0..n FilingPlan Eine unbestimmte Anzahl an Aktenplan-Objekten
Beispiel VB	<pre> Dim doc as New DOCUMENTS Try Dim filingPlans() As FilingPlan = Nothing filingPlans = doc.getFilingPlans() For Each fpa As FilingPlan In filingPlans Console.WriteLine("fp: {0} {1}", fpa.name, fpa.label) Next Return True Catch ex As Exception Console.WriteLine("getFilingPlans: {0}", ex.Message) Return False End Try </pre>
Erläuterung	Auf einem DOCUMENTS-Objekt werden Aktenpläne ermittelt und Name sowie Label auf der Konsole ausgegeben.
Kommentar	Die Methode steht erst ab der WSDL DOCUMENTS-4.0.1827 zur Verfügung

Documents.getFilingPlanXML

Beschreibung	Ermittelt eine XML-Beschreibung eines konkreten.
Eingabeparameter	name 1 String Technischer Bezeichner des Aktenplanes, für den die XML-Datei ermittelt werden soll.
Ausgabeparameter	Keine
Rückgabewert	1 String FilingPlan Die XML-Beschreibung in UTF-8 Encoding für den jeweiligen Aktenplan Aktenplans nach dem Schema aus dem Abschnitt 5.7.
Beispiel VB	<pre> Dim doc as New DOCUMENTS Try Dim filingPlans() As FilingPlan = Nothing filingPlans = doc.getFilingPlans() For Each fpa As FilingPlan In filingPlans Dim xml As New String(String.Empty) xml = doc.getFilingPlanXML(fpa.name) Console.WriteLine("FilingPlan: {0} ", fpa.name) Console.WriteLine(xml) Next Catch ex As Exception Console.WriteLine("Message: {0}", ex.Message) End Try </pre>
Erläuterung	Es werden die technischen Bezeichner aller Aktenpläne ermittelt und dann die jeweils zugehörige XML-Darstellung ausgelesen.
Kommentar	Die Methode steht erst ab der WSDL DOCUMENTS-4.0.1827 zur Verfügung

Documents.getFolderStructure

Beschreibung	Die Ordnerstruktur innerhalb eines Standardordners auflisten.
Eingabeparameter	folderType : 1 String Der Typbezeichner eines persönlichen Standardordners, für den die Struktur ermittelt werden soll. Eine Liste der möglichen Ordnerarten befindet sich im Anhang Bezeichner für persönliche Standardordner.
Ausgabeparameter	Keiner
Rückgabewert	baseFolder : 1 FolderDescription
Beispiel VB	<pre> Dim fd As FolderDescription fd = doc.getFolderStructure("Favourites") If fd.hasSubFolders Then For Each fod As FolderDescription In fd.subFolders Console.WriteLine("Subfolder: {0}", fod.label) Next Else Console.WriteLine("No subfolders") End If </pre>
Erläuterung	Auf einem gegebenen DOCUMENTS-Objekt wird für den „Favoriten“-Ordner eine Beschreibung der Ordnerstruktur angefordert. Verfügt der Ordner über weitere Unterordner (hasSubFolders), so werden ihre Bezeichnungen auf der Konsole herausgeschrieben.
Kommentar	Die Funktion garantiert nicht, dass die Ausgabereihenfolge der Ordner genau der Darstellung auf der Weboberfläche entspricht.

Documents.getInbox

Beschreibung	Ermittelt die Mappen und Statusinformationen dieser Mappen, die sich in der Inbox des eingeloggten Benutzers befinden.
Eingabeparameter	Keiner
Ausgabeparameter	fileStatuses : 0..n

	FileStatus Eine unbestimmte Anzahl von
--	---

	FileStatus-Objekten, mit Informationen zu MappenId, Workflowschritt und Versendestatus.
Rückgabewert	idFolder: 1 String Die Id der Inbox.
Beispiel VB	<pre> Try Dim fileStatuses() As FileStatus = Nothing Dim idFolder As New String(String.Empty) idFolder = doc.getInbox(fileStatuses) ' How many files? Console.WriteLine("Inbox countFiles: {0}", fileStatuses.Length()) Return True Catch ex As Exception Console.WriteLine("getInbox Message: {0}", ex.Message) Return False End Try </pre>
Erläuterung	Auf einem DOCUMENTS-Objekt wird die Inbox eines bereits angemeldeten Benutzers geholt und die Mappenanzahl auf der Konsole ausgegeben.
Kommentar	

Documents.getMonitor

Beschreibung	Ermittelt die Versendungshistorie (Workflowschritte) zu einer Mappe.
Eingabeparameter	fileId: Die Id der Mappe, deren Monitoreinträge geholt werden soll.
Ausgabeparameter	Keiner
Rückgabewert	monitorEntries: 0..n MonitorEntry Eine unbestimmte Anzahl von Monitoreinträgen, die jeweils einen Schritt in der Versendungshistorie widerspiegeln.
Beispiel VB	<pre> Dim monitorentries() As MonitorEntry = Nothing monitorentries = doc.getMonitor(theFileId) 'iterates through all MonitorEntries For Each moe As MonitorEntry In monitorentries 'Print out the information of each entry Console.WriteLine() Console.WriteLine("Status: {0} fileOK: {1}", moe.status, moe.fileOk) Console.WriteLine("Executive: {0}", moe.executive) Console.WriteLine("EntryDate: {0} ResponseDate: {1}", moe.entryDate, moe.responseDate) Console.WriteLine("Task: {0} Comment: {1}", moe.task, moe.comment) Next </pre>
Erläuterung	Das DOCUMENTS-Objekt und die MappenId müssen bereits deklariert und instanziiert bzw. bekannt sein. Auf dem doc-Objekt wird die getMonitor-Methode aufgerufen und durch das zurückgegebene MonitorEntry-Array iteriert, wobei für jeden

	Eintrag die jeweiligen Eigenschaften auf der Konsole ausgegeben werden.
Kommentar	siehe auch Documents.getWorkflowSteps

Documents.getProperty

Beschreibung	Die Methode liest Eigenschaften aus, die auf dem Mappentypen einer Mappe hinterlegt wurden.
Eingabeparameter	fileId : 1 String Der technische Bezeichner der Mappe propertyNames : 0..n String Array mit den Namen der Eigenschaften, die für die Mappe ausgelesen werden sollen. Die Eigenschaftsnamen müssen mit einem führenden \$ beginnen.
Ausgabeparameter	
Rückgabewert	propertyValues : 0..n String Zu den Namen korrespondierendes String-Array mit den ermittelten Werten.
Beispiel VB	<pre> Dim idFolder As New String(String.Empty) idFolder = doc.getInbox(fileStatuses) ' Iterate through all inbox files If fileStatuses.Length > 0 Then For Each fs As FileStatus In fileStatuses Dim propertyNames As New List(Of String) propertyNames.Add("\$Test") Dim propertyValues() As String propertyValues = Nothing propertyValues = doc.getProperty(fs.fileId, propertyNames.ToArray) Dim i As Integer = 0 For Each val As String In propertyValues Console.WriteLine("Name: {0} Value: {1}", propertyNames.Item(i), val) i = i + 1 Next Next End If </pre>
Erläuterung	Es wird die Inbox des Benutzers ermittelt und auf jeder dieser Mappen der Wert der Eigenschaft (Property) "Test" ermittelt und auf der Kommandozeile angezeigt.
Kommentar	

Documents.getSentFolder

Beschreibung	Ermittelt die Mappen und Statusinformationen dieser Mappen, die sich im SentFolder (Gesendet) des eingeloggten Benutzers befinden.
Eingabeparameter	Keine
Ausgabeparameter	fileStatuses : 0..n

	FileStatus-Objekten, mit Informationen zu MappenId, Workflowschritt und Versendestatus.
Rückgabewert	idFolder: 1 String Die Id des SentFolder.
Beispiel VB	<pre>Try Dim fileStatuses() As FileStatus = Nothing Dim idFolder As New String(String.Empty) idFolder = doc.getSentFolder(fileStatuses) For Each fs As FileStatus In fileStatuses Console.WriteLine("FileStatusFileId: {0} IdWorkflowStep: {1} Status: {2}", fs.fileId, fs.idWorkflowStep, fs.status) Next Return True Catch ex As Exception Console.WriteLine("getSentFolder Message: {0}", ex.Message) Return False End Try</pre>
Erläuterung	Auf einem DOCUMENTS-Objekt wird der SentFolder (Gesendet) des aktuellen Benutzers geholt und für jedes im Rückgabe-Array befindliche Objekt die FileStatus-Informationen angezeigt.
Kommentar	

Documents.getTasks

Beschreibung	Ermittelt die Aufgaben zu einer Mappe.
Eingabeparameter	fileId: 1String Der technischen Bezeichner der Mappe.
Ausgabeparameter	Keine
Rückgabewert	tasks: 0..n String Die Aufgaben zu dieser Mappe.
Beispiel VB	<pre>Try Dim fileStatuses() As FileStatus = Nothing</pre>

	<pre> Dim idFolder As New String(String.Empty) ' Get the sentfolder idFolder = doc.getSentFolder(fileStatuses) ' Now iterate through that array For Each fs As FileStatus In fileStatuses Dim tasks() As String = Nothing ' get the tasks for this file tasks = doc.getTasks(fs.fileId) If tasks.Length > 0 Then For Each s As String In tasks Console.WriteLine("Task: {0}", s) Next End If Next Return True Catch ex As Exception Console.WriteLine("GetTask Message: {0}", ex.Message) Return False End Try </pre>
Erläuterung	Zunächst holen wir uns den SentFolder des Benutzers. Aus den zurückgegebenen FileStatus-Objekten wird die Id der jeweiligen Mappe ermittelt und mit Hilfe dieser Id werden dann die Aufgaben der Mappe geholt, die als ein String-Array vorliegen. Wenn in dem Array mindestens ein Eintrag existiert, so wird durch das ganze Array iteriert und die einzelnen Aufgaben nacheinander auf der Konsole ausgegeben.
Kommentar	

Documents.getWorkflowPattern

Beschreibung	Ermittelt die im System vorhandenen Workflows.
Eingabeparameter	Keiner
Ausgabeparameter	Keiner
Rückgabewert	workflowPattern: 0..n WorkflowPattern Eine unbestimmte Anzahl von WorkflowPattern-Objekten mit Informationen zu Id und Namen des Workflows.
Beispiel VB	<pre> Try ' here we store the returned WorkflowPattern Dim wps() As WorkflowPattern 'fetch them wps = doc.getWorkflowPattern() 'iterate through all workflows For Each wp As WorkflowPattern In wps ' and print the name Console.WriteLine("Name: {0}", wp.nameWorkflowPattern) Next Return True Catch ex As Exception </pre>

	<pre> Console.WriteLine("WorkflowPattern Message: {0}", ex.Message) Return False End Try </pre>
Erläuterung	Für einen bereits angemeldeten Benutzer wird auf einem DOCUMENTS-Objekt die Methode zur Ermittlung der Workflows aufgerufen. In dem zurückgegebenen Array von WorkflowPattern-Objekten werden für jedes Objekt der Name und die Id auf der Konsole ausgegeben.
Kommentar	

Documents.getWorkflowSteps

Beschreibung	Ermittelt die Workflowschritte, die an der angegebenen Mappe hängen.
Eingabeparameter	fileId : 1String Der technischen Bezeichner (Id) der Mappe.
Ausgabeparameter	Keine
Rückgabewert	WorkflowSteps: 0..n WorkflowStep
Beispiel VB	<pre> Dim fileStatuses() As FileStatus = Nothing doc.getInbox(fileStatuses) For Each fs As FileStatus In fileStatuses Dim wfsteps() As WorkflowStep ' get the workflowid and get the Steps wfsteps = doc.getWorkflowSteps(fs.fileId) For Each wstep As WorkflowStep In wfsteps Console.WriteLine("Entry: {0}", wstep.entryDate) Console.WriteLine("Finish: {0}", wstep.finishDate) Next Next </pre>
Erläuterung	Es wird die Inbox des Benutzers ausgelesen und anhand der Mappen-Id wird für jede Mappe die Versendehistorie ausgelesen und dabei für jeden Workflowschritt die Eingangs- und Ausgangszeit auf der Konsole ausgegeben.
Kommentar	siehe auch Documents.getMonitor

Documents.listPossibleActions

Beschreibung	Ermittelt die gegenwärtig möglichen Workflowschritte einer Mappe.
Eingabeparameter	fileId : 1 String Die Id der Mappe.
Ausgabeparameter	Keine
Rückgabewert	workflowActions: 0..n WorkflowAction Eine unbestimmte Anzahl von WorkflowAction -Objekten mit Informationen zu Id und Bezeichnung der möglichen Benutzeraktion.
Beispiel VB	<pre> doc.getInbox(fileStatuses) ' Now iterate through all inbox files If fileStatuses.Length > 0 Then For Each fs As FileStatus In fileStatuses Console.WriteLine(New String("-", 10)) Next End If </pre>

	<pre> Console.WriteLine("fileId : {0}", fs.fileId) Dim workflowActions() As WorkflowAction = Nothing ' get all actions workflowActions = doc.listPossibleActions(fs.fileId) For Each wa As WorkflowAction In workflowActions ' and print them Console.WriteLine("Label: {0} Id: {1}", wa.label, wa.id) Next Next Else Console.WriteLine("No files found") End If </pre>
Erläuterung	Es wird die Inbox des angemeldeten Benutzers ausgelesen und für jede Mappe ein Array mit den möglichen Aktionen geholt (workflowActions()). Für jede dieser Aktionen werden dann auf der Konsole die Id und ihre Bezeichnung ausgegeben.
Kommentar	

Documents.listPublicFolders

Beschreibung	Ermittelt die öffentlichen Ordner.
Eingabeparameter	typeOption: 1 Integer mit Wert 0 Der Parameter ist für einen zukünftigen Gebrauch bestimmt und muss vorerst mit dem Wert 0 belegt werden. sortOption: 1 Integer mit Wert 0 Der Parameter ist für einen zukünftigen Gebrauch bestimmt und muss vorerst mit dem Wert 0 belegt werden..
Ausgabeparameter	Keine
Rückgabewert	folders: 0..n FolderDescription2
Beispiel VB	<pre> Dim fd() As FolderDescription2 fd = doc.listPublicFolders(typeOption, sortOption) For Each fd2 As FolderDescription2 In fd Console.WriteLine("{0} {1}", fd2.label, fd2.name) Next ' fd2 </pre>
Erläuterung	Der Code ermittelt die oberste Ebene in der Struktur der öffentlichen Ordner.
Kommentar	Es werden nur die Ordner zurückgegeben, die für den angemeldeten Benutzer im Web sichtbar sind. Ordner, die nicht angezeigt werden und nicht freigegeben wurden, werden nicht mit aufgeführt.

Documents.login

Beschreibung	Führt eine Anmeldung über den Documentsproxy aus.
Eingabeparameter	user: String Der Benutzername, unter dem die Anmeldung ausgeführt werden soll. principal: String Der Mandant, für den der Benutzer angemeldet werden soll.

	passwd: String Das Passwort, mit dem sich der Benutzer beim Proxy authentifiziert. code: String/ Leerstring Reserviert für zukünftige Erweiterungen. In dieser Version ist ein Leerstring zu übergeben. locale: String Kürzel für die Portalsprache „de“, „en“..., in der ergonomische Bezeichner, Fehlermeldungen u.a.m. ausgegeben werden.
Ausgabeparameter	Keine
Rückgabewert	session: 1 String Der Rückgabewert enthält dieSessionID, die dem Benutzer im Proxy zugeordnet ist.
Beispiel VB	<pre>Dim doc as New DOCUMENTS Dim session As New String(String.Empty) Try session = doc.login("schreiber","peachit","willi","", "de") Catch ex As Exception Console.WriteLine("Login Message: {0}", ex.Message) End Try</pre>
Erläuterung	Auf einem DOCUMENTS-Objekt wird die login-Methode aufgerufen und die ermittelte Session in einem String gespeichert. Bei einer ggfs. auftretenden Exception wird der Text der Exception auf der Konsole ausgegeben.
Kommentar	Zu Beginn der Anwendung muss ein login erfolgen. Ein logout braucht erst am Ende der Anwendung zu erfolgen, ist also nicht bei jedem einzelnen Funktionsaufruf notwendig. Sie sollten das Logout in einem Finally-Block ausführen, damit auch nach einem Catch sichergestellt ist, das ein Logout erfolgt. Ansonsten gilt der Benutzer bis zum Timeout im Proxy als angemeldet, was eine erneute Anmeldung verhindert. Wurde ein Login durchgeführt, so muss auch ein Logout erfolgen, da die Session im Proxy sonst nicht unmittelbar freigegeben werden kann. Konkurrierende Anmeldungen beim Proxy, d.h. zwei oder mehrere Anmeldungen für denselben Benutzer werden nicht unterstützt. Kann die Anmeldung nicht durchgeführt werden, da der Benutzer/Mandant nicht bekannt ist, wird ein Fehler ausgelöst. Soll der User unter einem anderen Account angemeldet werden, ist die Funktion Documents.trustedLogin zu verwenden.

Documents.logout

Beschreibung	Führt eine Abmeldung für den angemeldeten Benutzer aus.
Eingabeparameter	Keiner
Ausgabeparameter	Keiner
Rückgabewert	Keiner
Beispiel VB	<pre>Dim doc as New DOCUMENTS Dim session As New String(String.Empty) Try doc.logout() Catch ex As Exception Console.WriteLine("Login Message: {0}", ex.Message)</pre>

	End Try
Erläuterung	Auf einem DOCUMENTS-Objekt wird die logout-Methode aufgerufen. Bei einer ggfs. auftretenden Exception wird der Text der Exception auf der Konsole ausgegeben.
Kommentar	Wurde ein Login durchgeführt, so muss auch ein Logout erfolgen, da die Session im Proxy sonst nicht unmittelbar freigegeben wird. Siehe auch Documents.login.

Documents.report

Beschreibung	Suche nach Dokumenten eines Mappentyps und Ausgabe von Feldwerten als HTML-Tabelle.
Eingabeparameter	filetypes: 0..n String Eine Liste von Mappentypnamen, alternativ zu filetype verwendbar. archives: 0..n String Eine Liste von Suchquellen-Ids der Archive (siehe 5.2). Allerdings werden keine Archive bei dieser Funktion unterstützt, daher soll diese List immer leer lassen. filter: 1 String Eine Suchbedingung, wie im Anhang unter Syntaxbeschreibung für Filterausdrücke beschrieben; kann leer bleiben, um alle Mappen des übergebenen Typs anzuzeigen, soweit das verwendete Benutzerkonto dazu berechtigt ist. sort: 1 String Ein Sortierkriterium in der Schreibweise Feldname+ für aufsteigende bzw. Feldname- für absteigende Sortierung oder ein Leerstring. Derzeit kann nur nach einem Feld sortiert werden. Ferner muss das Feld, nach dem sortiert werden soll auch in der columns Auflistung vorhanden sein. columns: 0..n String Die Namen der auszugebenden Felder. Neben den Indexfeldern des Mappentyps können hier auch Mappenattribute, wie z.B. der Titel abgefragt werden. Siehe dazu Anhang 5.4 für spezielle Feldbezeichner.
Rückgabewert	report: String Es wird keine in sich abgeschlossene und somit gültige HTML-Seite ausgegeben, sondern nur ein <table> Element).
Beispiel VB	<pre>' The filetypes to be searched Dim filetypes As New List(Of String) filetypes.Add("ftEmployee") ' The fields whose values should be returned Dim columns As New List(Of String) columns.Add("hrLastName") Dim archives As New List(Of String) Dim filter As New String(String.Empty) filter = "hrLastName~'Sc*" "hrLastName~'Sc*" Dim sort As New String(String.Empty) ' sort in reverse alphabetic order</pre>

	' the field in the sort string has also to be in the columns-list sort = "hrLastName-" Dim report As New String(String.Empty) report = doc.report(filetypes.ToArray, archives.ToArray, columns.ToArray, filter, sort)
Erläuterung	Der Mappentyp "ftEmployee" wird nach allen Mitarbeitern durchsucht, in deren Nachnamen die Buchstabenkombination „sc“ auftaucht. Die Treffer werden in absteigender alphabetischer Reihenfolge – abhängig von der verwendeten Datenbank – in das table-Element herausgeschrieben und auf der Konsole angezeigt.
Kommentar	Wird der vom Server zurückgesendete String direkt geparkt, so sei darauf hingewiesen, dass durch die Einbettung in XML einige Zeichen des HTML-Reports umcodiert werden, z.B. „>“ in „&gt;“ usw. Den <tr>-Elementen des ausgegebenen Reports wird ein Parameter id hinzugefügt, der jeweils einen String enthält, welcher die gefundene Mappe eindeutig identifiziert. Zur Bildung der Filterausdrücke siehe im Anhang unter Syntaxbeschreibung für Filterausdrücke. Die Funktion kann nicht für Archivmappen verwendet werden.

Documents.report2

Beschreibung	Suche nach Mappen eines oder mehrerer Mappentypen/ Suchquellen-Ids für Archive (siehe 5.2) unter Ausgabe einer Trefferliste.
Eingabeparameter	filetypes: 0..n String Eine Liste von Mappentypnamen archives: 0..n String Eine Liste von Suchquellen-Ids der Archive (siehe 5.2).. EE.x-Archive werden nicht unterstützt. filter: 1 String Eine Suchbedingung, wie im Anhang 1 beschrieben; kann leer bleiben, um alle Mappen des übergebenen Typs anzuzeigen, soweit das verwendete Benutzerkonto dazu berechtigt ist. sort: 1 String Ein Sortierkriterium in der Schreibweise Feldname+ für aufsteigende bzw. Feldname- für absteigende Sortierung oder ein Leerstring. Derzeit kann nur nach einem Feld sortiert werden. Ferner muss das Sortierfeld auch beim Parameter columns vorhanden sein. columns: 0..n String Die Namen der auszugebenden Felder. Neben den Indexfeldern des Mappentyps können hier auch Mappenattribute, wie z.B. der Titel abgefragt werden. Letzteres geht über reservierte Bezeichner, die im Anhang 5.4 aufgelistet sind. Wird "Hit_Id" als Spalte hinzugefügt, so wird in dieser Spalte die Mappen-Id bzw. bei Archivmappen der <i>Archivmappen-Key</i> aufgeführt. Wird "Hit_ArchiveKey" als Spalte hinzugefügt, so wird in dieser Spalte nur bei Archivmappen der <i>Archivmappen-Key</i> aufgeführt.

Ausgabeparameter	Keiner
Rückgabewert	hitlist: 1 HitList Eine Trefferliste mit den Ergebniszeilen- und Spalten.
Beispiel Vb	<pre> Dim filetypes As New List(Of String) filetypes.Add("ftRecord") ' The fields whose values should be returned Dim columns As New List(Of String) columns.Add("DlcFile_Id") columns.Add("DlcFile_Created") Dim archives As New List(Of String) ' We are searching for files which are created today or yesterday Dim filter As New String(String.Empty) Dim today As String = DateTime.Now.ToShortDateString Dim yesterday As String = DateTime.Now.AddDays(- 1).ToShortDateString filter = "Search_DateFrom>=" + yesterday + " AND Search_DateUntil<=" + today + "" ' and sort them in order of their creation date Dim sort As New String(String.Empty) sort = "DlcFile_Created" 'search Dim hitl As New HitList hitl = doc.report2(filetypes.ToArray, archives.ToArray, columns.ToArray, filter, sort) Dim rows As Integer = hitl.rows Dim cols As Integer = hitl.columns If Not hitl Is Nothing Then Dim hds() As HitData = hitl.hit For i As Integer = 0 To rows - 1 Console.WriteLine(New String("-", 10)) Dim hd As HitData = hds(i) For j As Integer = 0 To cols - 1 Console.WriteLine("Hit {0} Col {1} Value {2}", i + 1, j + 1, hd.column(j)) Next Next i End If ' hitl </pre>

Erläuterung	Es wird der Mappentyp "ftRecord" nach den Mappen durchsucht, die gestern oder heute erstellt wurden. Dazu werden für den Filter die entsprechenden DateTime-Objekte erzeugt und im Filter eingesetzt. Die Sortierung der Treffer soll nach dem Entstehungs-Zeitstempel der Mappe erfolgen. Die resultierende Trefferliste wird zeilenweise durchgearbeitet und die Werte für die zwei angeforderten Spalten auf der Konsole ausgegeben.
Kommentar	Wegen früherer technischer Probleme bei der Ausgabe von Arrays mit null Elementen gibt die Funktion in Ihrer Version als report eventuell noch einen Leerstring anstelle eines Arrays aus, wenn keine Mappen gefunden wurden. Anwendungen sollten daher vor dem Zugriff auf den Report dessen Datentyp überprüfen. Zur Bildung der Filterausdrücke siehe im Anhang unter Syntaxbeschreibung für Filterausdrücke.

Documents.report3

Beschreibung	Suche nach Mappen eines oder mehrerer Mappentypen in einem oder mehrerer Archiven unter Berücksichtigung eines bestimmten Trefferlistenschemas.
Eingabeparameter	filetypes, filter, sort, columns: Siehe Documents.report2 Die zu ermittelnden columns müssen in dem Trefferlistenschema des Enterprise Archives angelegt sein. DlcFile_Id, Hit_ArchiveKey und Hit_Id können nur in den columns verwendet werden, nicht beim Parameter filter oder sort. archives: : 0..n String Eine Liste von Suchquellen-Id der Archive (siehe 5.2). hitlistname : 1..1 String Bei EASY-Archiven der Name des Trefferlistenschemas, das in den Archiven angelegt wurde. Bei EAS -Archiven darf hier kein Wert eingegeben werden.
Ausgabeparameter	Keiner
Rückgabewert	hitlist: 1 HitList Eine Trefferliste mit den Ergebniszeilen- und Spalten.

Beispiel Vb	<pre> Dim filetypes As New List(Of String) ' The fields whose values should be returned ' All of them has to be defined in the hitlistname Dim columns As New List(Of String) columns.Add("DlcFile_Id") columns.Add("DlcFile_Created") Dim archives As New List(Of String) archives.Add("ftRecord@peachitStore1") Dim hitlistname as String hitlistname = "" Dim filter As New String(String.Empty) Dim today As String = DateTime.Now.ToShortDateString Dim yesterday As String = DateTime.Now.AddDays(- 1).ToShortDateString filter = "Search_DateFrom>=" + yesterday + " AND Search_DateUntil<=" + today + "" ' and sort them in order of their creation date Dim sort As New String(String.Empty) sort = "DlcFile_Created" 'search Dim hitl As New HitList hitl = doc.report3(filetypes.ToArray, archives.ToArray, columns.ToArray, filter, sort) Dim rows As Integer = hitl.rows Dim cols As Integer = hitl.columns If Not hitl Is Nothing Then Dim hds() As HitData = hitl.hit For i As Integer = 0 To rows - 1 Console.WriteLine(New String("-", 10)) Dim hd As HitData = hds(i) For j As Integer = 0 To cols - 1 Console.WriteLine("Hit {0} Col {1} Value {2}", i + 1, j + 1, hd.column(j)) Next Next i End If ' htill </pre>
-------------	--

Erläuterung	Es wird der Mappentyp "ftRecord" in dem Archiv „ftRecord@peachitStore1“ nach den Mappen durchsucht, die gestern oder heute erstellt wurden. Dazu werden für den Filter die entsprechenden DateTime-Objekte erzeugt und im Filter eingesetzt. Die Sortierung der Treffer soll nach dem Entstehungs-Zeitstempel der Mappe erfolgen. Die resultierende Trefferliste wird zeilenweise durchgearbeitet und die Werte für die zwei angeforderten Spalten auf der Konsole ausgegeben.
Kommentar	Siehe Documents.report2 .

Documents.runScript

Beschreibung	Ein Portalscript ausführen.
Eingabeparameter	fileId : 1 String Der technische Bezeichner der an das Skript zu übergebenden Mappe. name : String Der Name des aufzurufenden Skripts. paramList : 0..n String Eingabeparameter für das Skript als Liste. Der Liste muss paarweise ein Parametername und dahinter der dazugehörige Wert hinzugefügt werden.
Ausgabeparameter	errorMsg : String Fehlertext bei fehlgeschlagenem Aufruf oder fehlerhafter Ausführung des Skripts. returnValue : 1 String Der Rückgabewert des Skripts.
Rückgabewert	returnStatus : Integer Bei erfolgreichem Aufruf der Operation ist dieser Wert 0, ansonsten ein Fehlercode.
Beispiel Vb	<pre> Dim fileStatuses() As FileStatus = Nothing doc.getInbox(fileStatuses) ' Now iterate through all inbox files If fileStatuses.Length <= 0 Then Console.WriteLine("No files found") Return False End If Dim fileId As New String(fileStatuses(0).fileId) Dim name As New String("test") Dim errorMsg As New String(String.Empty) Dim returnValue As New String(String.Empty) Dim paramList As New List(Of String) paramList.Add("msg") paramList.Add("This message should be returned") Dim retStatus As New Integer retStatus = doc.runScript(fileId, name, paramList.ToArray, errorMsg, returnValue) Console.WriteLine("fileId: {0} scriptname: {1}", fileId, name) Console.WriteLine("errorMsg: {0}", errorMsg) </pre>

	Console.WriteLine("retStatus: {0} returnValue: {1}", retVal, returnValue)
Erläuterung	Als Voraussetzung muss über den Client ein Skript „test“ mit einem Eingabeparameter „msg“ vom Typ String und der Voreinstellung „Message“ eingerichtet worden sein. Als Skriptcode ist „return msg;“ einzugeben. Aus der Inbox des Benutzers wird die fileId der ersten Mappe ermittelt. Anschließend werden die nötigen Daten für den Skriptaufruf zusammengestellt. Dabei wird auch eine Stringliste für die zu übergebenden Parameter aufgebaut, die die Einträge „msg“ und „This message should be returned“ besitzt. Die Ausgabeparameter und der retStatus werden nach Skriptaufruf auf der Konsole ausgegeben.
Kommentar	Verfügbar ab otrisPORTAL 5.0o bzw. ELC3.50o. Skriptparameter müssen derzeit immer als Strings übergeben werden. Ebenso wird der Rückgabewert des Skripts als String ausgegeben.

Documents.sendFileAdHoc

Beschreibung	Direktversendung einer Mappe
Eingabeparameter	fileId : String Der technische Bezeichner der Mappe receivers: 1..n String Die Namen der Benutzer oder auch Gruppen, an die die Mappe gesendet werden soll. Es muss mindestens ein Empfänger angegeben werden. sendMode: String Die Versendeart. Gültige Werte sind sequential(nacheinander) und parallel_info(gleichzeitig zur Information). task: String Eine Aufgabenbeschreibung für die Empfänger der Mappe. backWhenFinished: Boolean Gibt an, ob die Mappe nach dem Umlauf wieder an das eigene Benutzerkonto zurückgesendet werden soll.
Ausgabeparameter	Keiner
Rückgabewert	Keiner
Beispiel VB	<pre>Dim fileStatuses() As FileStatus = Nothing doc.getInbox(fileStatuses) If fileStatuses.Length <= 0 Then Console.WriteLine("No files found") Return False End If Dim fileId As New String(fileStatuses(0).fileId) Dim sendMode As New String("parallel_info") Dim task As New String("Info by sendfile") Dim backWhenFinished As Boolean = False</pre>

	<pre>' we send the file to the group Dim receivers As New List(Of String) receivers.Add("Mitarbeiter") ' send doc.sendFileAdHoc(fileId, receivers.ToArray, sendMode, task, backWhenFinished)</pre>
Erläuterung	Es wird die erste Mappe aus der Inbox des angemeldeten Benutzers genommen und an die Gruppe „Mitarbeiter“ parallel zur Information gesendet.

Documents.triggerAction

Beschreibung	Löst eine Benutzeraktion auf einer Mappe aus.
Eingabeparameter	fileId: 1 String Die Id der Mappe. actionId: 1 String Die Id der auszulösenden Aktion. comment: 1 String Der Bearbeitungskommentar, der in die Versendungshistorie eingetragen werden soll.
Ausgabeparameter	Keiner
Rückgabewert	Keiner
Beispiel VB	<pre>Dim fileStatuses() As FileStatus = Nothing doc.getInbox(fileStatuses) If fileStatuses.Length <= 0 Then Console.WriteLine("No files found") Return False End If Dim fileId As New String(fileStatuses(0).fileId) Dim workflowActions() As WorkflowAction = Nothing ' get all actions workflowActions = doc.listPossibleActions(fileId) For Each wa As WorkflowAction In workflowActions ' and print them Console.WriteLine("Label: {0} Id: {1}", wa.label, wa.id) doc.triggerAction(fileId, wa.id, "Performed TriggerAction") Console.WriteLine("Action performed: {0} ", wa.label) Exit For Next</pre>
Erläuterung	Es werden für die erste Mappe in der Inbox des Benutzers die möglichen Benutzeraktionen ermittelt. Die Angaben zur ersten Aktion werden auf der Konsole ausgegeben. Mit doc.TriggerAction wird diese Aktion dann auch auf einem DOCUMENTS-Objekt ausgelöst.
Kommentar	

Documents.startWorkflow

Beschreibung	Startet einen Workflow auf einer angelegten Mappe.
Eingabeparameter	fileId: 1 String Die Id der Mappe. idWorkflowPattern: 1 String

	Id des zu startenden Workflows.
Ausgabeparameter	Keiner
Rückgabewert	Keiner
Beispiel VB	<pre> ' Get the Inbox Dim fileStatuses() As FileStatus = Nothing doc.getInbox(fileStatuses) If fileStatuses.Length <= 0 Then Console.WriteLine("No files found") Return False End If ' And grab the id of the first file Dim fileId As New String(fileStatuses(0).fileId) Dim wps() As WorkflowPattern ' Get all Workflowpattern wps = doc.getWorkflowPattern() If wps.Length <= 0 Then Console.WriteLine("No workflow found") Return False End If ' And grab the id of the first workflowpattern Dim workflowId As New String(wps(0).idWorkflowPattern) ' Now start the workflow for this file doc.startWorkflow(fileId, workflowId) </pre>
Erläuterung	Auf einem DOCUMENTS-Objekt wird die Inbox des angemeldeten Benutzers ermittelt und die fileId der ersten Mappe ausgelesen. Anschließend wird die Id des ersten Workflows durch Abfrage der Workflowliste ermittelt und der Workflow auf der Mappe gestartet.
Kommentar	Zur Bestimmung der idWorkflowPattern siehe Documents.getWorkflowPattern.

Documents.testSession

Beschreibung	Prüft, ob für den aktuellen SOAP-Client eine Session besteht
Eingabeparameter	Keiner
Ausgabeparameter	Keiner
Rückgabewert	valid: 1 Boolean Zeigt, ob eine gültige Session für den Client eingerichtet wurde.
Beispiel VB	<pre> While(doc.testSession()) doc.logout() End While </pre>
Erläuterung	Auf einem DOCUMENTS-Object wird geprüft, ob für den Client eine Session zu dem SOAP-Proxy eingerichtet wurde. Ist dies der Fall, so wird der Client abgemeldet.
Kommentar	

Documents.trustedLogin

Beschreibung	Führt eine Anmeldung über den Documentsproxy aus, wobei der User unter einem anderen Account angemeldet wird.
Eingabeparameter	user: String

	Der Benutzername, für den die Anmeldung zu einem anderen Account durchgeführt werden soll. Für diesen Benutzer muss über den DOCUMENTS-Client, das Property (Eigenschaft) „trustedLoginAccount“ mit dem Wert „1“ vorhanden sein. principal: String Der Mandant, für den der Benutzer angemeldet werden soll. passwd: String Das Passwort, mit dem sich der Benutzer beim Proxy authentifiziert. asUser: String Der User, zu dem gewechselt werden soll. code: String/ Leerstring Reserviert für zukünftige Erweiterungen. In dieser Version ist ein Leerstring zu übergeben. locale: String Kürzel für die Portalsprache „de“, „en“..., in der ergonomische Bezeichner, Fehlermeldungen u.a.m. ausgegeben werden.
Ausgabeparameter	Keine
Rückgabewert	session: 1 String Der Rückgabewert enthält dieSessionID, die dem Benutzer im Proxy zugeordnet ist.
Beispiel VB	<pre>Dim doc as New DOCUMENTS Dim session As New String(String.Empty) Try ' 'schreiber' login as user 'oppen' session = doc.trustedLogin("schreiber","peachit","willi","oppen","", "de") Console.WriteLine("TrustedLogin Message: {0}",doc.UserInfo()) Catch ex As Exception Console.WriteLine("TrustedLogin Message: {0}", ex.Message) End Try</pre>
Erläuterung	Auf einem DOCUMENTS-Objekt wird die trustedLogin-Methode aufgerufen, in der User 'schreiber' als User 'oppen' angemeldet wird. Bei einer ggfs auftretenden Exception, z.B. wenn das Property 'trustedLoginAccount' beim User nicht vorhanden oder auf einen Wert ungleich '1' gesetzt ist, wird der Text der Exception auf der Konsole ausgegeben. In der Ausgabe der UserInfo-Methode sollte bei einem erfolgreichen Wechsel des Benutzers 'Bernhard Oppen' ausgegeben werden.
Kommentar	Es gelten dieselben Einschränkungen wie beimDocuments.login.

Documents.userInfo

Beschreibung	Ermittelt den vollständigen Namen des Benutzers, für den gegenwärtig die Session auf dem Proxy eingerichtet ist.
Eingabeparameter	Keiner
Ausgabeparameter	Keiner
Rückgabewert	name: 1 String

	Der vollständige Name des Benutzers, also Vor - und Nachname. Z.B. „Willi Schreiber“.
Beispiel VB	<pre>Dim doc as New DOCUMENTS Dim name as New String(String.Empty) Try name = doc.userInfo() Console.WriteLine("userInfo Name: {0} ", name) Catch ex As Exception Console.WriteLine("userInfo Message: {0}", ex.Message) End Try</pre>
Erläuterung	Auf einem DOCUMENTS-Objekt wird die userInfo-Methode aufgerufen, um den Namen des aktuell angemeldeten Benutzers zu ermitteln. Dieser wird dann auf der Konsole ausgegeben.
Kommentar	Ist kein Benutzer angemeldet und somit auch keine Session auf dem Proxy eingerichtet, so wird ein Fehler ausgelöst.

3.2 DOCUMENTS-Klassen

ArchiveStatus

Beschreibung	Enthält Information darüber, inwieweit die Archivierung einer Mappe erfolgreich war.
Member	fileId : 1 String die Id der Mappe, die im Archiv abgelegt werden sollte. messages : 1 String Ein Leerstring, falls die Mappe problemlos archiviert werden konnte; ansonsten ein Fehlerstring.
Kommentar	Die Fehlerstrings beginnen mit einer technischen Kurzbezeichnung, der durch Pipezeichen getrennt Zusatzangaben folgen können. Eine Sonderstellung nimmt dabei das Kürzel "DlcErrFieldsNotArchived" ein, welches besagt, das einige Mappenfelder in der Archivstruktur nicht existieren und daher weggelassen wurden. Alle anderen "DlcErr..."-Bezeichner lassen darauf schließen, dass die Mappe nicht im Archiv angekommen ist. Falls eine Mappe schon einmal archiviert war, wählt DOCUMENTS dasselbe Ziel-/Ursprungsarchiv. Ansonsten muss über den Portal-Manager dem jeweiligen Mappentyp ein "aktives Archiv" zugewiesen sein.
Wird verwendet in	Documents.archiveFiles

DeleteStatus

Beschreibung	Enthält Stausinformationen zu einem Löschvorgang.
Member	id : 1 String die Id des betreffenden Dokuments/Mappe. deleted : 1 Boolean Zeigt an, ob das Dokument/Mappe erfolgreich gelöscht wurde. messages : 1 String Falls bei einem Dokument/Mappe ein Fehler aufgetreten ist, wird hier ein Fehlerstring eingetragen. Die Fehlerstrings beginnen mit einer technischen Kurzbezeichnung, der durch Pipezeichen getrennt Zusatzangaben folgen können. Die Bezeichnungen beginnen immer mit der Zeichenfolge "DlcErr".

	Bei Fehlern auf Mappenebene werden keine identischen Fehlerstrings für jedes einzelne Dokument ausgegeben, sondern es wird eine Exception ausgelöst.
Wird verwendet in	Documents.deleteDocuments Documents.deleteFiles

DocInfo

Beschreibung	Hält die Informationen zu den Dokumenten, die einer Mappe angehängt sind.
Member	id: 1 String Die Id des Dokumentes. name: 1 String Der Dokumentname. comment: 1 String Kommentar zum Dokument. ab ELC 3.60f/ otris Portal 6.0f size: 1String Die Dokumentgröße mit Einheitskürzel.
Wird verwendet in	FileInfo

DocUploadData

Beschreibung	Hält Informationen zu einem hochzuladenen Dokument.
Member	name: 1 String Der Dokumentname (= Dateiname inklusive Erweiterung). register: 1 String Der Name eines Dokumentenregisters, dem das Dokument zugeordnet werden soll. Nur falls ein Standardregister definiert ist, darf dieser Parameter weggelassen werden. data: 1base64Binary Der Dateiinhalt als base64 codierter String. In VB genügt ein Byte Array zur Aufnahme des Dateiinhaltes. Die Konvertierung wird über SOAP automatisch vorgenommen.
Wird verwendet in	Documents.createFile Documents.editFile Documents.createFile2

DocUploadData2

Beschreibung	Hält Informationen zu einem hochzuladenen Dokument.
Member	name: 1 String Der Dokumentname (= Dateiname inklusive Erweiterung). register: 1 String Der Name eines Dokumentenregisters, dem das Dokument zugeordnet werden soll. Nur falls ein Standardregister definiert ist, darf dieser Parameter weggelassen werden. data: 1base64Binary Der Dateiinhalt als base64 codierter String. In VB genügt ein Byte Array zur Aufnahme des Dateiinhaltes. Die Konvertierung wird über SOAP automatisch vorgenommen. replace: 0..1Boolean

	Gibt an, ob ein Dokument in einer aktiven Mappe ggfs. durch das hochzuladene Dokument mit demselben Namen ersetzt werden soll. Der Parameter ist bei Archivmappen ignoriert. versioning: 0..1 Boolean Gibt an, ob ein Dokument in einer aktiven Mappe ggfs. durch das hinzufügende Dokument mit demselben Namen versioniert werden soll, falls die Ersetzung vorgesehen ist (replace = true). Der Parameter wird auch bei Archivmappen ignoriert.
Wird verwendet in	Documents.editFile2

DocUploadData3

Beschreibung	Hält Informationen zu einem hochzuladenen Dokument.
Member	name: 1 String Der Dokumentname (= Dateiname inklusive Erweiterung). register: 1 String Der Name eines Dokumentenregisters, dem das Dokument zugeordnet werden soll. Nur falls ein Standardregister definiert ist, darf dieser Parameter weggelassen werden. path: 1 String Der Pfad der hochzuladenen Datei relativ zum BlobBasePath aus docsoaproxy.ini. replace: 0..1 Boolean Gibt an, ob ein Dokument in einer aktiven Mappe ggfs. durch das hochzuladene Dokument mit demselben Namen ersetzt werden soll. Der Parameter wird bei Archivmappen und Documents.createFile3 ignoriert. versioning: 0..1 Boolean Gibt an, ob ein Dokument in einer aktiven Mappe ggfs. durch das hinzufügende Dokument mit demselben Namen versioniert werden soll, falls die Ersetzung vorgesehen ist (replace = true). Der Parameter ist auch bei Archivmappen und Documents.createFile3 ignoriert. deleteBlob: 0..1 Boolean Gibt an, ob die Datei nach dem Hochladen aus dem Dateisystem gelöscht wird.
Wird verwendet in	Documents.createFile3 Documents.editFile3

FieldData

Beschreibung	Hält Name und Wert zu einem Mappenfeld.
Member	name: 1 String Der Feldname value: 1 String Der Feldwert
Wird verwendet in	Documents.createFile Documents.createFile2 Documents.editFile Documents.getFileInfo FileInfo Documents.editFile2 Documents.createFile3

	Documents.editFile3
--	---------------------

FieldDescription

Beschreibung	Enthält Detailinformation zu den Feldern eines FileTypes.
Member	name: 1String Der ergonomische Feldname in der Locale des Benutzers. id: 1 String Der technische Name des Feldes. fieldlabel: 0..1 String Dient zur Ermittlung des vollständigen Labels des Felds und seines ergonomischen Namen in der eingestellten Sprache, indem man im „categories“-Parameter „fields,fieldlabel“ bzw „fields,fieldlabel.locale“ angibt. Falls für das Feld kein Label eingetragen ist, gibt das fieldlabel einen Leerstring für „fields,fieldlabel“ bzw. den technischen Namen des Felds für „fields,fieldlabel.locale“ zurück. Für weitere Informationen siehe Documents.describeFileType. type: 1 String Gibt an, zu welchem Typ das Feld im Documents gehört: • string: einzeliger Text • text: mehrzeiliger Text • boolean: Wahrheitswert • date: Datum • enum: Aufzählung siehe unten • numeric: Numerischer Wert • reference: Verknüpfungsfeld zu einer anderen Mappe • other: Keiner der oben genannten. enum: 0..n Strings Enthält die Ausprägungen des enum-Feldes, wenn das Feld vom Type „enum“ ist und der „categories“-Parameter „fields, enum“ enthält, siehe Documents.describeFileType. Dabei werden die Werte für die einzelnen Sprachen in einem String (1) dargestellt. „de:Jahr; en:Year“.
Wird verwendet in	FileTypeDescription Documents.describeFileType Documents.describeFileType2

FileInfo

Beschreibung	Enthält Detailinformation zu den Feldern eines FileTypes.
--------------	---

Member	idFile: 1 String Der eindeutige Bezeichner der zu zugrundeliegenden Mappe. filetypeId: 1 String Die Id des Mappentypen, zu dem die Mappe gehört. filetypeName: 1 String Der Name des Mappentypen für diese Mappe. fileTypeLabel: 1 String Der ergonomische Namen des Mappentypes auf der Weboberfläche. documents: 0..n DocInfo Die Dokumente der Mappe mit Informationen zu Größe, Name und Id. fieldvalues: 0..n FieldData Enthält die Name/Wert Paare zu den einzelnen Mappenfeldern.
Wird verwendet in	Documents.GetFilesInfo

FileStatus

Beschreibung	Hält Statusinformationen zu einer Mappe.
Member	fileId: String Die Id der Mappe, sie kann beispielsweise zur Ermittlung von Feldinhalten verwendet werden. idWorkflowStep: String Der aktuelle Workflowschritt, in dem sich die Mappe befindet. status: String Der Status, in dem sich die Mappe im System befindet. Mögliche Ausprägungen: ○ DlcFile_Status_Default Neuangelegte Mappe ○ DlcFile_Status_New Mappe ist aus Wiedervorlage in die Inbox gekommen ○ DlcFile_Status_ToForward Benutzer sperrt Mappe und muss diese weiterleiten ○ DlcFile_Status_Info Benutzer hat die Mappe zur Info erhalten ○ DlcFile_Status_Task Benutzer sperrt die Mappe, muss diese weiterleiten und die Mappe hat eine Aufgabe für den Benutzer ○ DlcFile_Status_CancelWorkflow Mappe wurde durch einen Workflow abgebrochen ○ DlcFile_Status_FileBack Mappe ist aus einer Versendung zurückgekommen ○ DlcFile_Status_ArchFile_Default Mappe ist Archivmappe ○ DlcFile_Status_Consultation Mappe mit Rückfrage eines anderen Benutzers ○ DlcFile_Status_Deleted Mappe ist gelöscht
Wird verwendet in	Documents.getInbox Documents.getSentFolder

FileTypeArchiveInfo

Beschreibung	Enthält identifizierende Daten zu den Archiven, auf denen sich ein filetype bezieht.
Member	nameDest: 1 String Name des Standard-Zielarchivs für den Mappentyp. Bei einem EAS der technische Bezeichner des Archivservers, z.B. „peachitStore1“ idDest: 1 String

	Bezeichner der dazugehörigen importierten Archivstruktur. Bei einem EAS-Archiv ein Leerstring. keyDest: 1 String Korrespondierender Key zum FileType, wie er im Archiv adressiert wird. Beim EAS-Archiv Mappentyp gefolgt von „@“ gefolgt von technischen Bezeichner des Archivservers, z.B. „ftRecord@peachitStore1“ arcSrc: 0..nSourceArchiv Liste der Ursprungsarchive, deren Mappen durch diesen Mappentypen wieder in aktive Vorgänge umgewandelt werden können.
Wird verwendet in	FileTypeDescription Documents.describeFileType Documents.describeFileType2

FileTypeDescription

Beschreibung	Enthält Detailinformationen zu einer Mappe.
Member	id: 0..1 String Die Id des FileTypes im System. name: 0..1 String Der technische Name des FileTypes. archiveinfo: 0..1 FileTypeArchiveInfo Informationen zu den Archiven, auf die sich der Mappentyp bezieht. docregisters: 0..nRegisterDescription Information zu den Dokumentenregistern. fields: 0..nFieldDescription Beschreibungen zu den einzelnen Feldern. workflowinfo: 0..1 FileTypeWorkflowInfo Informationen darüber welche Workflow diesem Mappentyp zugeordnet ist.
Wird verwendet in	Documents.describeFileType. Dabei entscheidet der „categories“-Parameter, welche Datenstrukturen aufgefüllt werden.

FileTypeDescription2

Beschreibung	Enthält Detailinformationen zu einer Mappe. Bis auf das <i>Label</i> ist die Datenstruktur mit FileTypeDescription identisch.
Member	id: 0..1 String Die Id des FileTypes im System. name: 0..1 String Der technische Name des FileTypes. label: 0..1 String Der Oberflächenbezeichner des FileTypes in der Sprache, in der der Benutzer angemeldet ist. archiveinfo: 0..1 FileTypeArchiveInfo Informationen zu den Archiven, auf die sich der Mappentyp bezieht. docregisters: 0..nRegisterDescription Information zu den Dokumentenregistern. fields: 0..nFieldDescription Beschreibungen zu den einzelnen Feldern. workflowinfo: 0..1 FileTypeWorkflowInfo

	Informationen darüber welche Workflow diesem Mappentyp zugeordnet ist.
Wird verwendet in	Documents.describeFileType2. Dabei entscheidet der „categories“-Parameter, welche Datenstrukturen aufgefüllt werden. Die Datenstruktur steht erst ab der WSDL DOCUMENTS-4.0.1827 zur Verfügung

FileTypeShortDescr

Beschreibung	Hält Statusinformationen zu einer Mappe.
Member	id: 1 String Die Id des Mappentypen im System. name: 1 String Der technische Name des FileTypes.
Wird verwendet in	Documents.getFileTypes

FileTypeWorkflowInfo

Beschreibung	Enthält Angaben zum Standardworkflow eines Mappentypen
Member	StdForwarding: 1 RegisterDescription Enthält eine Beschreibung der Standard-Versendeliste. Workflow: 1 WorkflowIdent Beschreibung des aktuellen Workflows, der dem Filetype zugeordnet ist.
Wird verwendet in	FileTypeDescription Documents.describeFileType Documents.describeFileType2

FilingPlan

Beschreibung	Enthält Angaben zu einem Aktenplan
Member	name: 1 String Der technische Bezeichner des Aktenplans. label: 1 String Bezeichner auf der Bneutzeroberfläche in der jeweils angemeldeten Sprache description: 1 String
Wird verwendet in	Documents.getFilingPlans

FolderDescription

Beschreibung	Beschreibt eine Ebene innerhalb eines Standard-Ordners
Member	id: 1 String Technischer Bezeichner des Ordners, der z.B. für Documents.BrowseFolderbenutzt werden kann. label: 1 String Ordnername auf der Weboberfläche. hasSubFolders: 1 Boolean Gibt an, ob dieser Ordner weitere Unterordner besitzt. subFolders: 0..n FolderDescription Eine unbestimmte Anzahl von Ordnerbeschreibungen der Unterordner, was ein rekursives Navigieren durch den Ordnerbaum erlaubt.
Wird verwendet in	Documents.getFolderStructure

FolderDescription2

Beschreibung	Beschreibt eine Ebene innerhalb eines Öffentlichen-Ordners
Member	id: 1 String Id des Ordners, die z.B. für Documents.BrowseFolder benutzt werden kann. folderType: 1 String mit Angabe des Ordner types • Public – Öffentlich • PublicDynamicFilter – Öffentlich mit dynamischen Filter • OnlySubFolder – Enthält nur Unterordner label: 1 String Ordnername auf der Weboberfläche. labelML: 1 String Das komplette mehrsprachige Label des Ordners. name: 1 String Technischer Name des Ordners hasSubFolders: 1 Boolean Gibt an, ob dieser Ordner weitere Unterordner besitzt. subFolders: 0..n FolderDescription2 Eine unbestimmte Anzahl von Ordnerbeschreibungen der Unterordner, was ein rekursives Navigieren durch den Ordnerbaum erlaubt.
Wird verwendet in	Documents.listPublicFolders

FolderFile

Beschreibung	Beschreibt eine Mappe innerhalb eines Standard-Ordners.
Member	id: 1 String Id der Mappe innerhalb des Folders. values: 1 StringList Die Werte der Felder für diese Mappe, sofern diese angefordert wurden.
Wird verwendet in	Documents.getFolderStructure Documents.browseFolder

HitData

Beschreibung	Enthält die Daten für eine Trefferliste, die bei einer Suche erzeugt wurde.
Member	column: 0..n String Die Spalten in dem ausgegebenen Report. Diese kann kleiner sein als in der Documents.report2 -Methode mit columns vorgegeben wurde, falls ein angefordertes Feld im Mappentyp nicht existiert oder für das verwendete Benutzerkonto generell nicht lesbar ist. reserved: 1 Integer Reserviert für zukünftige Funktionalität.
Wird verwendet in	HitList Documents.report2

HitList

Beschreibung	Enthält die Daten für eine Trefferliste, die bei einer Suche erzeugt wurde.
Member	rows: 1 Integer

	Die Zeilenanzahl der Trefferliste, entspricht der Trefferanzahl. columns: 1 Integer Die Spaltenzahl in dem ausgegebenen Report. Diese kann kleiner sein als durch <i>fields</i> vorgegeben, falls ein angefordertes Feld im Mappentyp nicht existiert oder für das verwendete Benutzerkonto generell nicht lesbar ist. hit: 0..n HitData. Enthält die eigentlichen Trefferwerte.
Wird verwendet in	Documents.report2 Documents.report3

MonitorEntry

Beschreibung	Enthält die Daten für den Monitoreintrag in einer Mappe.
Member	status: 1 String Der Mappenstatus, wie folgt: • "Angelegt" "Created" • "Wartend" "Waiting" • "Gesperrt" "Locked" • "Weitergeleitet" "Forwarded" • "Durchgeführt" "Processed" • "Informiert" "Informed" • "Beendet" "Finished" • "Zurückgeholt" "Canceled" • "Initiator wartend" "Waiting Initiator" • "Beendet und zurück" "Finished and Back" • "Versendet" "Sent" • "Gelesen" "Read" • "Nicht berücksichtigt" "Not considered" • "Eskalation 1" "Escalation 1" • "Eskalation 2" "Escalation 2" • "Eskalation" "Escalation" • "Fehlgeschlagen" "Failed" • "Angezeigt" "Shown" executive: 1 String Das Login des Mappenbearbeiters. entryDate: 1 String Datum und Uhrzeit des Mappeneingangs. responseDate: 1 String Datum und Uhrzeit der Mappenweiterleitung. fileOk: 1 Boolean Ob die Mappe im „OK“-Status weitergeleitet wurde. task: 1 String Die Aufgabe, die der Mappe bei Mappeneingang für die Bearbeitung zugewiesen wurde. comment: 1 String Der Kommentar mit dem die Weiterleitung versehen wurde.
Wird verwendet in	Documents.getMonitor

RegisterDescription

Beschreibung	Enthält Angaben zur Identifizierung eines Dokumentenregisters.
Member	name: 1 String Enthält den Namen des Dokumentenregisters.

	id: 1 String Die Id des Dokumentenregisters.
Wird verwendet in	FileTypeDescription Documents.describeFileType Documents.describeFileType2

SourceArchiv

Beschreibung	Enthält Angaben zur Identifizierung eines Ursprungarchives.
Member	nameSrc: 1String Name des Ursprungarchives idSrc: 1String Id des Ursprungarchives keySrc: 1 String Schlüssel des Ursprungarchives, wie er im Archiv vergeben wurde
Wird verwendet in	FileTypeDescription Documents.describeFileType. Documents.describeFileType2

StringList

Beschreibung	Enthält eine unbestimmte Anzahl von Strings.
Member	string: 0..n Strings
Wird verwendet in	Documents.browseFolder
Kommentar	Je nach verwendeter Programmiersprache und IDE können String-Arrays noch einmal in einer Stringliste gekapselt sein.

WorkflowAction

Beschreibung	Enthält Id und Label einer Aktion in einem Workflow.
Member	id: 1 String Die Id der Aktion. label: 1 String Das Label der Aktion auf der Weboberfläche.
Wird verwendet in	Documents.listPossibleActions

WorkflowIdent

Beschreibung	Enthält Version und Id zum aktuellen Workflow eines Mappentypen.
Member	version: 1 String Versionsbezeichnung des Workflows. id: 1 String Id des Workflows.
Wird verwendet in	FileTypeDescription Documents.describeFileType Documents.describeFileType2

WorkflowPattern

Beschreibung	Enthält Name und Id eines Workflows.
Member	idWorkflowPattern: 1 String Id des Workflows. nameWorkflowPattern: 1 String Der Workflowname.
Wird verwendet in	Documents.getWorkflowPattern

WorkflowStep

Beschreibung	
Member	comment: 1 String Der Kommentar der dem Übergang von einem Workflowschritt zu nächsten mitgegeben wurde. entryDate: 1 String Das Datum und Uhrzeit des Mappeneingangs ("Erhalten am") fileOK: 1 String Kennzeichnet, ob die Mappe im Zustand "OK" weitergeleitet wurde. finishDate: 1 String Datum und Uhrzeit des Mappenausgangs ("Reaktion am") groupFlag: 1 Boolean Kennzeichnet, ob dieser Workflowschritt einer Gruppe zugewiesen wurde. hasAgent: 1 Boolean Kennzeichnet, ob dieser Workflowschritt durch einen Vertreter bearbeitet wurde. idWorkflowStep: 1 String Die Id des Workflowschrittes locksFile: 1 Boolean Kennzeichnet, ob dieser Workflowschritt die Mappe exklusiv zum Bearbeiten sperrt. redirectionDescription: 1 String Enthält eine Kurzinfo darüber, wer welchen Mitarbeiter bei der Bearbeitung vertreten hat. status: 1 String Der Status, in dem sich Mappe auf dem Workflowschritt befindet. Siehe MonitorEntry/status task: 1 String Die Aufgabe, die dem Bearbeiter des Workflowschrittes zugewiesen wurde. user: 1 String Das login des Bearbeiters . Dies kann auch ein Skript oder bei einem automatischen Ablauf z.B. auch eine "Entscheidung" sein. wasRedirected: 1 Boolean Kennzeichnet, ob dieser Workflowschritt auf einer Rückfrage beruht.
Wird verwendet in	Documents.getWorkflowSteps siehe auch Documents.getMonitor

3.3 Weitere Funktionen

WSDL

Beschreibung	WSDL durch einen Http-Request vom aktiven Proxy auslesen.
Beispiel VB	<pre> Try Dim url As String url = "/?wsdl" Dim webRequest As System.Net.HttpWebRequest webRequest = Nothing Dim webResponse As System.Net.HttpWebResponse </pre>

	<pre> webResponse = Nothing ' doc.Url Proxy location and port webRequest = CType(System.Net.WebRequest.Create(doc.Url + url), System.Net.HttpWebRequest) webResponse = CType(webRequest.GetResponse(), System.Net.HttpWebResponse) Dim receiveStream As System.IO.Stream = webResponse.GetResponseStream() Dim encode As System.Text.Encoding = System.Text.Encoding.GetEncoding(0) Dim readStream As New System.IO.StreamReader(receiveStream, encode) Console.WriteLine("Response stream received") Dim read(256) As [Char] Dim count As Integer = readStream.Read(read, 0, 256) Console.WriteLine("WSDL ..." + System.Environment.NewLine) While count > 0 Dim str As New [String](read, 0, count) Console.WriteLine(str) count = readStream.Read(read, 0, 256) End While Console.WriteLine("... WSDL") readStream.Close() webResponse.Close() Catch ex As Exception Console.WriteLine("sampleGetWSDL Message: {0}", ex.Message) End Try </pre>
Erläuterung	Wir ermitteln über einen Http-Get Aufruf die aktuelle WSDL vom Proxy

4. HTTPS-Unterstützung (SSL/TLS)

Ab **DOCUMENTS** 4.0d #1870 unterstützt der SOAPProxy alternativ zu HTTP-Verbindungen auch HTTPS-Verbindungen (TLS1.2+) zum SOAP-Client.

Die SOAPProxy-seitige HTTPS-Unterstützung beschränkt sich ausschließlich auf eine Transportverschlüsselung. Eine Client-Authentifizierung durch den SOAPProxy wird NICHT unterstützt.

Diese Dokumentation setzt Kenntnisse in der Funktionsweise von SSL/TLS und der Verwendung von digitalen Zertifikaten (X.509) voraus. Dargestellt werden hier die Konfigurationen des SOAPProxys und Codebeispiele für die Verwendung.

4.1 Konfigurationen in der docsoaproxy.ini

Standardmäßig ist die HTTPS-Unterstützung ausgeschaltet. Soll eine verschlüsselte Verbindung verwendet werden, müssen die folgenden Optionen in der INI-Datei `/soaproxy/docsoaproxy.ini` konfiguriert werden:

- `SSL=1`
aktiviert die HTTPS-Unterstützung
- `keyfile=server.pem`
absoluter Pfad oder zum Installationsverzeichnis relativer Pfad zu einer Zertifikats-/Schlüssel-Datei.
Die pem-Datei enthält den privaten RSA-Schlüssel und das Zertifikat des SOAPProxys ggfs. gefolgt von den Zwischenzertifikaten einer Zertifikatskette.

Der private Schlüssel muss im PEM Format (Base64-kodiert) umschlossen von
-----BEGIN ENCRYPTED PRIVATE KEY-----
und
-----END ENCRYPTED PRIVATE KEY-----
eingefügt werden. Die einzelnen Zertifikate werden ebenfalls im PEM Format jeweils umschlossen von
-----BEGIN CERTIFICATE-----
und
-----END CERTIFICATE-----
eingefügt.

- `Keypasswd=geheim`
Das Kennwort zum Lesen des privaten Schlüssels in dem „keyfile“. Das Kennwort muss im Klartext angegeben werden.

Wenn auf der SOAP-Client Seite eine Serverauthentifizierung durchgeführt werden soll, dann benötigt er dazu die entsprechenden vertrauenswürdigen CA-Stammzertifikate.

Abhängig von der verwendeten Client-Technologie werden unterschiedliche Zertifikatspeicher verwendet. Ein mittels .net erstellter Client unter Windows verwendet implizit den

Zertifikatspeicher des Betriebssystems. Das entsprechende Verwaltungswerkzeug öffnet Windows z.B. mit dem Kommandozeilenaufruf „MMC certmgr.msc“. Dagegen braucht ein gSOAP-Client eine eigene Stammzertifikatedatei (siehe 4.3 für mehr Informationen). Die Clientanwendung muss sich hierbei selbst um die Bereitstellung und um die regelmäßige Aktualisierung der CA-Stammzertifikate kümmern.

Wenn es einem Angreifer gelingen sollte, dem Client ein gefälschtes Stammzertifikat unterzuschieben, oder wenn der Client einem kompromittierten Stammzertifikat vertraut, wäre die Verbindung nicht mehr sicher.

4.2 Erstellen eines selbst signierten X.509-Zertifikats

Für simple Testzwecke kann man ein selbst signiertes Zertifikat erstellen. Dafür braucht man nur OpenSSL auf dem Server installiert zu haben (z.B. zusammen mit EDA -> C:\Program Files\Documents4\eas\http\bin\openssl.exe). Mit dem folgenden Befehl wird ein 2048 Bit langer RSA-Schlüssel generiert und als Datei testkey.pem gespeichert. Ein selbst signiertes Zertifikat wird ebenfalls erstellt und in testcert.pem abgelegt. Das Zertifikat ist 365 Tage gültig.

```
openssl req -x509 -days 365 -newkey rsa:2048 -out testcert.pem -keyout testkey.pem
```

Unter Windows kann es erforderlich sein, die Konfigurationsdatei (z.B. C:\Program Files\Documents4\eas\http\conf\openssl.cnf) explizit mit -config <Pfad-auf-openssl.cnf> anzugeben:

```
openssl req -x509 -days 365 -newkey rsa:2048 -out testcert.pem -keyout testkey.pem -config <Pfad-auf-openssl.cnf>
```

Während der Erstellung werden noch einige Details des Zertifikats abgefragt. Nicht benötigte Felder kann man durch Eingabe eines Punktes "." überspringen:

```
writing new private key to 'testkey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
Country Name (2 letter code) [AU]:DE
State or Province Name (full name) [Some-State]:NRW
Locality Name (eg, city) []:Dortmund
Organization Name (eg, company) [Internet Widgits Pty Ltd]:CompanyName
Organizational Unit Name (eg, section) []:IT
Common Name (e.g. server FQDN or YOUR name) []:localhost
Email Address []:test@domain.de
```

Die Inhalte von testkey.pem und testcert.pem bilden aneinandergehängt den Inhalt der neuen server.pem (keyfile). Das oben angegebene Passwort steht für die Option keypasswd. Zur Verifizierung des Servers an der Client-Seite wird die Datei testcert.pem verwendet.

4.3 HTTPS/SSL fähiger gSOAP-Client mit Visual C++ 2008®

Im Abschnitt 2.7 haben wir beschrieben, wie man einen gSOAP-Client mit Visual C++ 2008® erstellt. Nun möchten wir ihn in einen HTTPS fähigen Client umwandeln (siehe [Secure SOAP Clients with HTTPS/SSL](#) aus dem gSOAP Benutzerhandbuch für mehr Informationen). Dafür setzen wir voraus, dass OpenSSL auf dem Client verfügbar ist. Für das Projekt sind die folgenden Einstellungen durchzuführen:

- Fügen Sie den Include-Pfad von OpenSSL (z.B. C:\openssl-win32\include) dem Projekt unter Projekt > Eigenschaften > C/C++ > Allgemein > Zusätzliche Includeverzeichnisse hinzu.
- Tragen Sie die Option -DWITH_OPENSSL unter Projekt > Eigenschaften > C/C++ > Befehlszeile > Zusätzliche Optionen ein.
- Fügen Sie den Lib-Pfad von OpenSSL (z.B. C:\openssl-win32\lib) unter Projekt > Eigenschaften > Linker > Allgemein > Zusätzliche Bibliotheksverzeichnisse hinzu.
- Tragen Sie ssl32.lib und libeay32.lib unter Projekt > Eigenschaften > Linker > Eingabe > Zusätzliche Abhängigkeiten ein.

Nun passen wir den Code wie folgt an.

```
int _tmain(int argc, _TCHAR* argv[])
{
    // Create a gSOAP client object.
    DOCUMENTS doc;

    // Endpoint URL of soapproxy (change as needed)
    doc.endpoint = "https://localhost:11001";

    // Init SSL
    soap_ssl_init();

    // Init gSOAP context
    if( soap_ssl_client_context(doc.soap,
        SOAP_SSL_DEFAULT, //flags for SSL client/server authentication settings
        NULL, //keyfile (cert+key): required only when client must
                authenticate to server
        NULL, //password to read the key
        "testcert.pem", //cafile to store trusted root certificates
        NULL, //capath to directory with trusted root certificates
        NULL //if randfile!=NULL: use a file with random data to seed
                randomness
    ) )
    {
        soap_print_fault(doc.soap, stderr);
        exit(1);
    }

    // Sample for login
    // ...
}
```

```
    return 0;
}
```

Dabei haben wir das in 4.2 erzeugte selbst signierte Zertifikat `testcert.pem` verwendet.

Schließlich kopieren wir die beiden DLLs `ssleay32.dll` und `libeay32.dll` in das Verzeichnis, wo die Anwendung (z.B. `docsoapsample.exe`) sich befindet. Es müssen genau die DLLs benutzt werden, die zu den eingebundenen Importbibliotheken passen. Anderenfalls wird die Anwendung wahrscheinlich abstürzen.

4.4 Weitere HTTPS/SSL fähige Clients

Wie oben bereits erwähnt wurde, verwendet ein mittels .net erstellter Client implizit die CA-Stammzertifikate aus dem Windows-Zertifikatspeicher. Für Testzwecke verwenden wir hier das Zertifikat `testcert.pem` aus 4.2 und importieren es mit dem Zertifikat-Snap-In (`certmgr.msc`) wie folgt:

- Klicken Sie auf den Ordner **Vertrauenswürdige Stammzertifizierungsstellen**, in den wir es importieren möchten!
- Wählen Sie den Menüpunkt **Aktion > Alle Aufgaben > Importieren**
- Klicken Sie „Weiter“ und folgen Sie den Anweisungen!

WCF-Client mit Visual C#

Der gemäß Abschnitt 2.8 erstellte WCF-Client mit Visual C# lässt sich nicht in einen HTTPS fähigen umwandeln, er erwartet „http“ als URI-Schema. Jedoch kann man einen HTTPS fähigen WCF-Client mit Visual C# auf dieselbe Weise erzeugen. Das Einzige, was wir vorher tun müssen, ist das URI-Schema „http“ der Location in der `DOCUMENTS.wsdl` auf „https“ zu ändern:

```
<SOAP:address location="https://localhost:11001"/>
```

Danachfolgen die Schritte aus dem Abschnitt 2.8.

VB-Client

Den VB-Client aus dem Abschnitt 0 kann man leicht in einen HTTPS fähigen Client umwandeln, indem man die URL des SOAPProxys von HTTP auf HTTPS ändert.

```
Dim doc as New DOCUMENTS

' Die URL unter der der Proxy für diese Anwendung erreichbar ist.

doc.Url = "https://localhost:11001"
```

PortalScripting-Client

Der JavaScript-Client aus der Dokumentation zum PortalScripting lässt sich so wie der VB-Client in einen HTTPS fähigen Client umwandeln.

```
// URL for the SOAP Proxy
var url = "https://localhost:11001/";
```

5. Anhang

5.1 Archivmappen-Key

5.1.1 EE.i:

Unter dem *Archiv-Key* versteht man den eindeutigen und identifizierenden Bezeichner eines Archives.

Format:

```
$ (#Lagerort)\Archivname@technischer Name des Archivservers im DOCUMENTS
```

z.B.:

```
$ (#STANDARD)\EINRECH@enterprisei
```

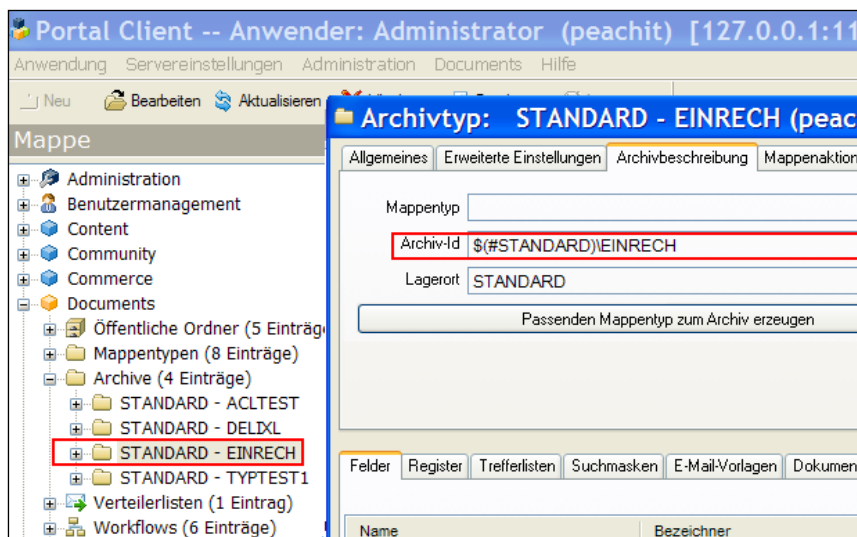


Abb. 11: EE.i Archiv-Key mit Detaildialog bei importierter Archivstruktur

Der *Archivmappen-Key* ist ein eindeutiger Bezeichner für eine Mappe innerhalb eines Archives. In diesen Keys sind manche Komponenten doppelt enthalten. Das liegt daran, dass DOCUMENTS den Teil vor dem ,|' als reine Mappenbezeichnung und den Teil dahinter zur Identifikation der Archivstruktur verwendet.

Format:

```
$ (#Lagerort)\Archivname,MappenId,VersionsId| $ (#Lagerort)\Archivname@technischer Name des Archivservers
```

z.B.:

```
$ (#STANDARD)\EINRECH,00000009,001| $ (#STANDARD)\EINRECH@eei1
```

5.1.2 EE.x:

Unter dem *View-Key* versteht man den eindeutigen und identifizierenden Bezeichner einer View (Sicht), die innerhalb eines Archives mit mindestens einem Schema verknüpft ist.

Format:

```
Unit=UnitName/Instance=InstanceName/View=ViewName@technischer Name des Archivservers im DOCUMENTS
```

z.B.:

```
Unit=Default/Instance=Default/View=EASY@enterprisex
```

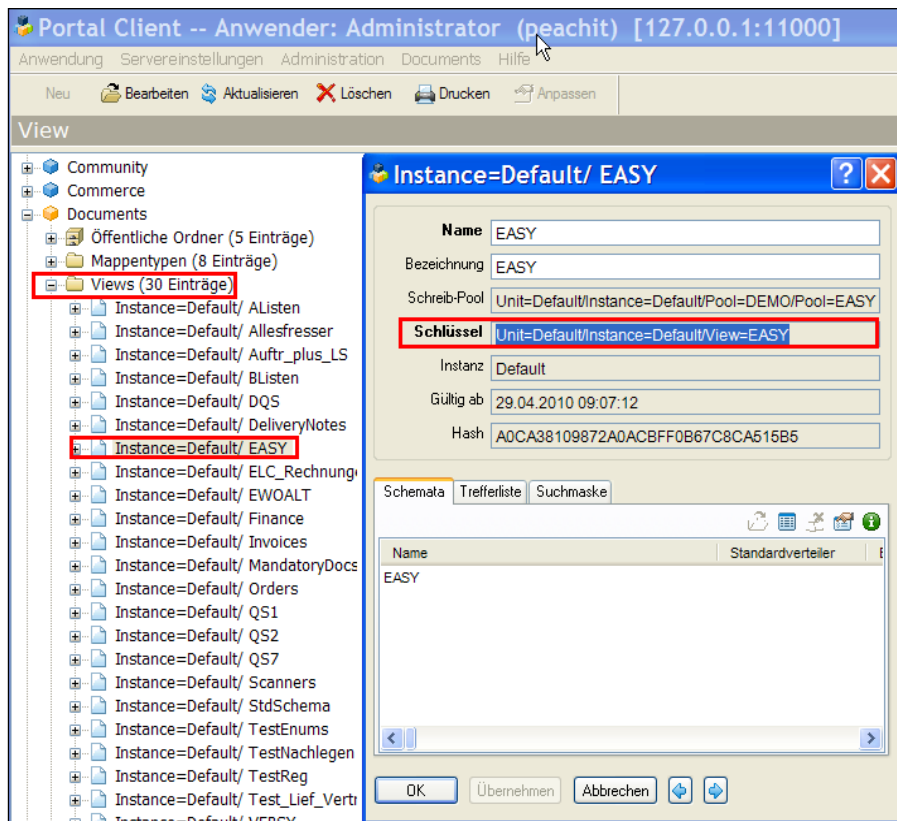


Abb. 12: EE.x View-Key mit Detaildialog bei importierter Archivstruktur

Der *Schema-Key* ist der identifizierende eindeutiger Bezeichner für ein Schema innerhalb eines Archives.

Format:

```
Unit=UnitName/Instance=InstanceName/DocumentSchema=SchemaName
```

z.B.:

```
Unit=Default/Instance=Default/DocumentSchema=EASY
```

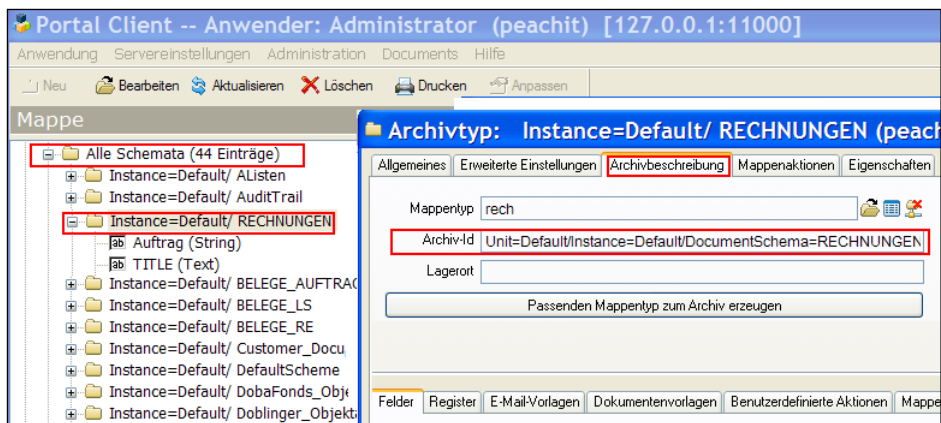


Abb. 13: EE.x Schema-Key mit Detaildialog bei importierter Archivstruktur

Der Archivmappen-Key einer EE.x-Mappe bildet sich wie folgt:

Format:

```
Unit=UnitName/Instance=InstanceName/Pool=PoolName  
/Document=MappenId|Schema-Key@technischer Name des Archivservers
```

z.B.:

```
Unit=Default/Instance=Default/Pool=DEMO/Pool=EASY/Document=EASY.45C76F221D3E11DF92AF080027B  
22D11|Unit=Default/Instance=Default/DocumentSchema=EASY@eex1
```

5.1.3 EAS und EBIS:

Ein Archiv-Key für EAS bzw. EBIS wird folgendermaßen zusammengesetzt:

```
Mappentyp@technischer Name des Archivservers im DOCUMENTS
```

z.B.:

```
ftRecord@peachitStore1
```

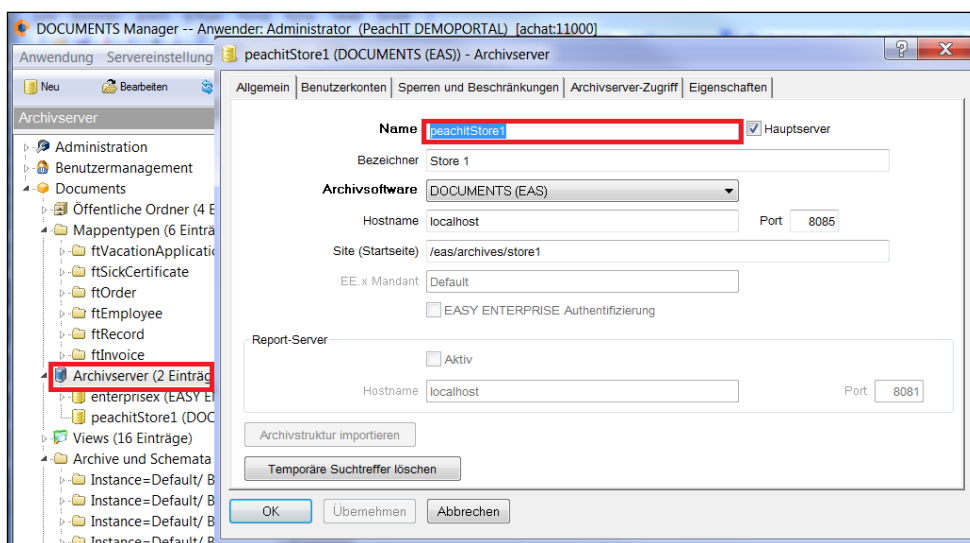


Abb. 14: Archivserver EAS

Der Archivmappen-Key einer EAS-Mappe bildet sich wie folgt:

Format:

Archivmappen-GUID Mappentyp@technischer Name des Archivservers
--

z.B.:

f5b2d475-587e-4020-bf15-7017722db169 ftRecord@peachitStore1

5.2 Suchquellen-Id eines Archives

Archivsoftware	Suchquellen-Id
EAS	Mappentyp@Archivserver
	z.B. Filetype1@store1
EE.i	\$(#Lagerort)\Archivname@Archivserver
	z.B. \$(#STANDARD)\REGTEST@eei1
EE.x	View-Key@Archivserver
	z.B. Unit=Default/Instance=Default/View=REGTEST@eex1

Die Suchquellen-Ids (siehe auch Archivmappen-Key) werden bei Documents.report2 und Documents.report3 verwendet.

5.3 Archivzielbezeichner

Archivsoftware	Archivzielbezeichner
EAS	Mappentyp@Archivserver
	z.B. Filetype1@store1
EE.i	\$(#Lagerort)\Archivname@Archivserver
	z.B. \$(#STANDARD)\REGTEST@eei1
EE.x	View-Key*Schema-Key@Archivserver
	z.B. Unit=Default/Instance=Default/View=REGTEST*Unit=Default/Instance=Default/DocumentSchema=REGTEST@eex1

Die Archivzielbezeichner (siehe auch Archivmappen-Key) werden bei den folgenden Funktionen verwendet:

- Documents.createFile
- Documents.createFile2
- Documents.createFile3
- Documents.editFile
- Documents.editFile2
- Documents.editFile3

5.4 Feldbezeichner für die Ausgabe von Mappenattributen in Reports/Suche

Es gibt zwei Kategorien von Spezialbezeichnern, die sich als Feldnamen zur Ausgabe von Mappenattributen verwenden lassen. Die Bezeichner der ersten Kategorie sind allgemein verwendbar. Es könnte jedoch aus technischen Gründen in späteren Versionen irgendwann einmal nötig werden, diese Bezeichner zu ändern.

Bezeichner	Bedeutung
DlcFile_Title	Mappentitel
DlcFile_Created	Erstelldatum
DlcFile_Owner	Eigentümer der Mappe
DlcFile_LastModified	Datum der letzten Bearbeitung
DlcFile_LastEditor	letzter Bearbeiter
DlcFile_Id	eindeutiger technischer Bezeichner der Mappe

Die zweite Kategorie von Spezialbezeichnern wird sich nicht mehr ändern. Wenn jedoch zu einem dieser Bezeichner ein gleichnamiges Feld existiert, dann wird für den betroffenen Mappentyp der entsprechende Feldwert anstelle des gleichnamigen Mappenattributs ausgegeben.

Bezeichner	Bedeutung
Title	Mappentitel
CreatedAt	Erstelldatum
FileOwner	Eigentümer der Mappe
LastModifiedAt	Datum der letzten Bearbeitung
LastEditor	letzter Bearbeiter
id	eindeutiger technischer Bezeichner der Mappe

5.5 Bezeichner für persönliche Standardordner

Bezeichner	Bedeutung
Favourites	Favoritenordner
Inbox	Eingangsordner
Sent	Gesendet
SendingFinished	Ordner "Versendung beendet"
InWork	Ordner "In Arbeit"
FollowUp	Wiedervorlageordner
Deleted	Ordner "Gelöscht"
Tasks	Aufgabenordner
LastUsed	Ordner "Zuletzt benutzt"
InTrouble	Ordner "InTrouble"
Wird verwendet in	Documents.getFolderStructure Documents.browseFolder

5.6 Syntaxbeschreibung für Filterausdrücke

Die allgemeine Schreibweise für einzelne Filterbedingungen bei den Operationen Documents.report und Documents.report2 lautet:

Feldname Vergleichsoperator Wert

Dabei sind folgende *Vergleichsoperatoren* erlaubt: = (gleich), < (kleiner), > (größer), <= (kleiner oder gleich), >= (größer oder gleich), ~ (enthält) und <> bzw. != (ungleich).

Vor und hinter dem Vergleichsoperator sollten keine Leerzeichen stehen. Der Wert muss in Hochkommata eingeschlossen werden, falls er Leerzeichen enthält.

Die erweiterte Suche in **DOCUMENTS** arbeitet außer bei Bereichssuchen immer mit dem Operator `~` (enthält). Nur bei diesem Operator wird der Suchbegriff in einzelne Wörter zerlegt. Alle anderen Operatoren verarbeiten den Suchbegriff als Ganzes.

Beispiel:

- Die Bedingung `Sachbearbeiter='Eva Frisch'` wird nur von Mappen erfüllt, bei denen das Feld `Sachbearbeiter` exakt den Text `Eva Frisch` ohne weitere Zeichen davor oder dahinter enthält.
- Dagegen wird die Bedingung `Sachbearbeiter~'Eva Frisch'` als `'Eva&Frisch'` (logisches Und) oder als `'Eva|Frisch'` (logisches Oder) interpretiert. Das hängt von der Serverkonfiguration ab (Siehe Parameter `LogicalAndBetweenSearchPhrases` in der „documents.ini“). Entsprechend werden mit Suchmethode `0` jene Mappen gefunden, die entweder beide Zeichenfolgen an beliebigen Positionen des Feldes enthalten (Und-Einstellung) oder auch nur eine davon (Oder-Einstellung).
- Das Rechercheergebnis ist hier dasselbe wie bei der Eingabe der beiden Wörter (ohne Hochkommata) im DOCUMENTS-Webformular für die erweiterte Suche.

Wenn nach Mappen gesucht werden soll, bei denen Eva Frisch oder Stefan Gross als Sachbearbeiter eingetragen sind, dann müsste die Filterbedingung folgendermaßen lauten:

```
Sachbearbeiter~'"Eva Frisch"|"Stefan Gross"'
```

Mehrere Filterbedingungen lassen sich zurzeit nur mit **AND** (logisches „und“) verknüpfen. Oder-Verknüpfungen sind auf dieser Ebene noch nicht möglich.

Erweitere Mappenattribute können in den Filterausdrücken wie folgt verwendet werden:

Bezeichner	Bedeutung
"Search_Fulltext"	Volltextsuche
"Search_DateFrom"	Von
"Search_DateUntil"	Bis
"DlcFile_Title"	Mappentitel
"Search_Owner"	Mappenbesitzer
"Search_LastEditor"	Letzter Bearbeiter
"Search_ModDateFrom"	Von (Datum der letzten Bearbeitung)
Search_ModDateUntil	Bis (Datum der letzten Bearbeitung)

Für weitere Details zum Thema Filterausdrücke wie z.B. archivspezifische Einschränkungen lesen Sie bitte auch das Kapitel „Using filter expressions with FileResultSets“ im *PortalScripting* Referenzhandbuch!

5.7 Schema für GetFilingPlanXML-Ausgabe

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">  
<xsd:element name="FilingPlan" type="FilingPlanEntryList"/>
```

```

<xsd:complexType name="FilingPlanEntryList">
    <xsd:sequence>
        <xsd:element name="FilingPlanEntry" type="FilingPlanEntryType"/>
    </xsd:sequence>
    <xsd:attribute name="name" type="xsd:string"/>
    <xsd:attribute name="label" type="xsd:string"/>
</xsd:complexType>
<xsd:complexType name="FilingPlanEntryType">
    <xsd:sequence>
        <xsd:element name="Index" type="xsd:string"/>
        <xsd:element name="Label" type="xsd:string"/>
        <xsd:element name="Custom1" type="xsd:string"/>
        <xsd:element name="Custom2" type="xsd:string"/>
        <xsd:element name="Custom3" type="xsd:string"/>
        <xsd:element name="FilingPlanEntry" type="FilingPlanEntryType"/>
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:string"/>
</xsd:complexType>
</xsd:schema>

```

5.8 Client-Timeouts

SOAP-Clients implementieren oft eine maximale Wartezeit auf die Antwort (Response) des SOAPProxys. Bei Zeitüberschreitung wird die Verbindung abgebrochen. Das Auftreten solch eines Client-Timeouts führt allerdings zum Session-Verlust. Genauer gesagt ist das Logout danach nicht mehr möglich, da dem Client hinterher eine gültige Session-Id fehlt. Das Sequenzdiagramm aus Abb. 15 veranschaulicht das Auftreten eines Client-Timeouts nach 60s.

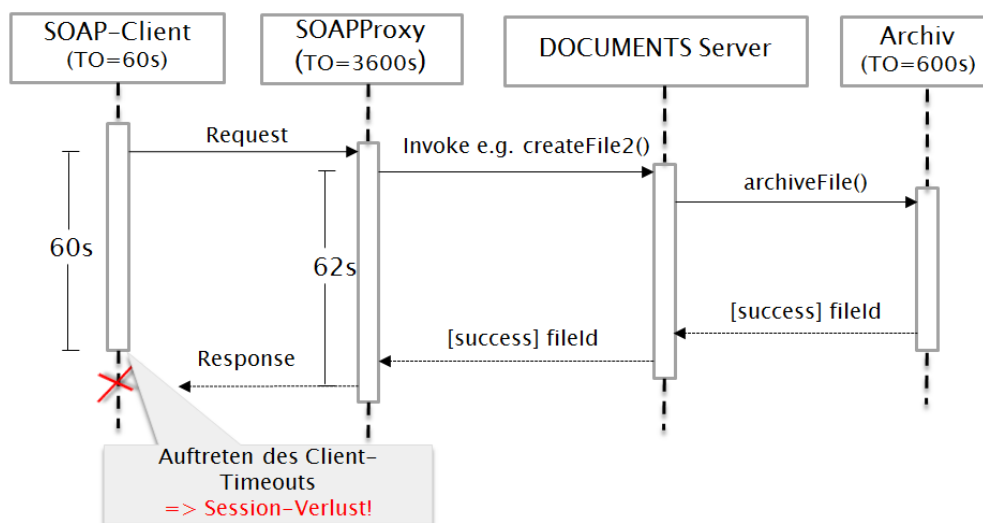


Abb. 15: Sequenzdiagramm eines Client-Timeouts

Die aus Clientsicht verlorene Session lebt im Proxy und im Server weiter. Wenn dies oft passiert, sind nach einer Weile alle Lizenzen belegt (max. 3 SOAP-Sessions pro Benutzer), und weitere Anmeldungen werden zurückgewiesen. Erst wenn für eine verlorene Session auch die Zeitspanne bis zum Proxy-Timeout abgelaufen ist (siehe Abb. 3), wird sie wieder frei. Um dieses Szenario zu vermeiden sollte das Zeitintervall für Client-Timeouts großzügig bemessen werden, sofern nicht darauf verzichtet werden kann.

6. Index

!=	99	Bis	99
~	99	boolean	79
<	99	browseFolder	28
<=	99	cancelWorkflow	30
<>	99	categories	38
=	99	describeFileType2	40
>	99	chcp	15
>=	99	code	64, 75
65001	15	Codepage	15
Abmeldung	48, 65	column	84
actionId	73	columns	66, 67, 85
addDocs	30, 32, 33, 42, 44, 46	comment	
Aktenplan	55	DocInfo	77
allAttributes	51, 53	followUp	47
allFields	51, 53	MonitorEntry	85
AND	99	triggerAction	73
Anmeldung	65	WorkflowStep	87
Archiv	76	count	28
Key	94	CreatedAt	98
archiveFiles	27	createFile	30, 77, 78
archiveinfo	39, 82	createFile2	31, 32
describeFileType2	41	data	50, 77, 78
FileTypeDescription2	82	date	79
archives	66, 67, 69	Dateiname	50
ArchiveStatus	76	Datentyp	10
Archivmappen	42, 43, 44, 45, 46, 94, 97	deleted	76
deleteFiles	36, 37	Deleted	98
getDocument	49	deleteDocuments	34
getFileInfo	51	deleteFiles	36
Key		deleteOnSuccess	27
EAS	96, 97	DeleteStatus	76
EE.i	94	describeFileType	38
EE.x	96	describeFileType2	40
Archivmappen-Key	44, 46	description	39, 41, 83
editFile	42	direktEAS	31, 32
arcSrc	82	DlcErr	29
asUser	75	DlcFile_Created	98
Aufgabenordner	98	DlcFile_Id	98
Ausprägung	79	DlcFile_LastEditor	98
Autotext	49	DlcFile_LastModified	98
autoTextNames	49	DlcFile_Owner	98
autoTextValues	49	DlcFile_Title	98, 99
backWhenFinished	72	docId	50
base64Binary	77, 78	docIds	34
baseFolder	56	DocInfo	77
Bearbeiter	98, 99	docregisters	39, 82
Bearbeitung	98	describeFileType2	41
Bearbeitungskommentar	73	documents	52, 80
Bedingung	99	DOCUMENTS.vb	10
Benutzeraktion	73	DocumentsHost	7
Benutzerrechte	54	DocumentsPort	7
Benutzers	75	DocUploadData	77

DocUploadData2.....	77, 78	listPossibleActions	63
Dokumente.....	51, 53	runScript.....	71
Download.....	49	sendFile	72
hinzufügen.....	30, 32, 33, 42, 44, 46	triggerAction.....	73
löschen	34	fileIds	
Dokumentenregisters	77, 78, 85	archivieren	27
Download.....	49	deleteFiles.....	36, 37
EAS.....	94	fileinfo	53
Ebene	83, 84	FileInfo.....	79
EE.i	94	fileId	80
EE.x	94	fileOk	85
Eigentümer.....	98	fileOK.....	87
Eingangskorb	47	FileOwner	98
Eingangsordner	98	files.....	28
Encoding	7	FileStatus	81
enthält.....	99	fileStatuses.....	56, 59
entryDate	85, 87	filetype.....	44, 46
enum.....	38, 79	createFile	30, 31, 32
describeFileType2	40	editFile.....	42
Ergebniszeilen	67, 69	FileTypeArchiveInfo	81
errorMsg	71	FileTypeDescription.....	82
Erstelldatum	98	FileTypeDescription2.....	82
executive.....	85	filetypeld	52, 80
failOnAllErrors	53	fileTypeLabel.....	52, 80
Favoritenordner.....	98	fileName	80
Favourites.....	98	filetypes	66, 67
Fehlerstring	76	FileTypeShortDescr.....	83
Fehlerstrings	29	FileTypeWorkflowInfo	83
Feldbezeichner	98	fileTypeName	52
Feldname		filter	66, 67
Sortierung	66	Filterausdrücke	66, 67, 69, 98
Feldwerte.....	51, 53	Filterbedingung.....	99
FieldData	78	Finally.....	65
FieldDescription	79	finishDate	87
fieldlabel	38, 79	FolderDescription	83
describeFileType2	40	FolderDescription2	83
fieldNames	28	FolderFile	84
fields	30, 31, 33, 38, 42, 44, 46, 82	folderId	28
describeFileType2	40	folderName	28
FileTypeDescription2.....	82	folders	64
fieldvalues	52, 80	folderType.....	28, 56
fileId.....	44, 46, 59, 73	followUp.....	47
ArchivStatus	76	FollowUp	98
cancelWorkflow	30	Funktionsumfang	4
createFile	30, 32, 33	Gelöscht.....	98
deleteDocument.....	34	Gesendet.....	98
editFile	42	getAutoText	49
FileInfo.....	51, 53	getDocument	49
FileStatus	81	getFileId	50
followUp.....	47, 48	getFileInfo	51
getAutotext.....	49	getFilesInfo	53, 80
getDokument	49	getFileTypes	54, 83
getMonitor	58	getFilingPlans.....	55
getTasks	61	getFilingPlanXML.....	55
getWorkflowSteps.....	63	getFolderStructure.....	55

getInbox.....	56, 81
getMonitor	58
getProperty.....	59
getSentFolder	59, 81
getTasks.....	61
getWorkflowPattern	62
getWorkflowSteps.....	63
gleich	99
größer	99
größer oder gleich	99
groupFlag.....	87
hasAgent.....	87
hasSubFolders	83, 84
HasValue	48
headline	28
hit	85
Hit-ArchiveKey	67
HitData	84
Hit-Id	67
hitlist.....	67, 69
HitList.....	84
hitlistname	69
Hochkommata	99
HTTP GET	87
id	98
DeleteStatus	76
describeFileType	38
describeFileType2	40
DocInfo	77
FieldDescription	79
FolderDescription	83
FolderDescription2	84
FolderFile	84
Mappentyp	82, 83
FileTypeDescription2.....	82
RegisterDSdescription	85
WorkflowAction	86
WorkflowIdent	86
idDest.....	81
idFolder	61
getInbox	57
idSrc.....	86
idWorkflowPattern	74, 86
idWorkflowStep	81, 87
ignoreRights.....	54
In Arbeit	98
Inbox.....	56, 98
initialValue	39
describeFileType2	41
InTrouble.....	98
InWork.....	98
isUTC	47
Key	
Archivmappen	
EAS	96, 97

EE.i	94
EE.x	96
Schema	95
View.....	95
keyDest.....	81
keyfield	42, 44, 46
keySrc	86
keyvalue.....	42, 44, 46
kleiner.....	99
kleiner oder gleich	99
Kommunikation.....	12
Konkurrierende Anmeldungen.....	65
Konsolenanwendung	14
Konsolenschriftart.....	15
Konvertierung	77, 78
label.....	83, 84, 86
FileTypeDescription2	82
LastEditor	98
LastModifiedAt	98
LastUsed	98
Leerzeichen.....	99
listPossibleActions	63
listPublicFolders	64
locale	65, 75
locksFile.....	87
login	64, 74
logout	48, 65
Löschvorgangs	34, 36, 37
Mappe	
identifizieren	43, 45
löschen	36, 37
Mappenattributen.....	98
Mappenbesitzer	99
Mappenstatus.....	85, 87
Mappentitel.....	98, 99
Mappentyp	
Detailinformation	38, 40
Mappentypen	54
messages	76
mime	50
Mimetype	50
monitorEntries	58
MonitorEntry	85
name.....	77, 78, 83, 84
describeFileType	38
describeFileType2	40
DocInfo.....	77
DocUploadData	77
FieldData	78
FieldDescription	79
FileTypeDescription	82
FileTypeDescription2	82
FileTypeShortDescr	83
getDocument	50
RegisterDescription	85

runScript	71	Search_LastEditor	99
userInfo	75	Search_ModDateFrom	99
nameDest	81	Search_ModDateUntil	99
Namen	75	Search_Owner	99
nameSrc	86	sendFileAdHoc	72
nameWorkflowPattern	86	SendingFinished	98
newId		sendMode	72
editFile	43	Sent	98
editFile2	44, 46	SequenceSessionId	7
nextIndex	28	session	65, 75
Nullable	48	Session	12, 74, 75
numeric	79	SessionId	7
Oder	99	size	50, 77
Öffentliche Ordner	84	Skript	71
Ordner		soapproxy	5
Mappen auflisten	28	SOAP-Schicht	11
Ordnerstruktur	56	SOAP-Session	12, 74
other	79	sort	66, 67
paramList	71	sortOption	64
passwd	64, 75	SourceArchiv	86
Portalscript	71	Spezialbezeichnen	98
PortalServerEncoding	7	Standardordner	29
preview	28	Standardordners	56
previousIndex	28	startIndex	28
principal	64, 75	Startobjekt	14
Projektmappenexplorer	13	startWorkflow	73
propertyNames	59	status	34, 81, 85, 87
propertyValues	59	statusList	27, 36, 37
Proxy	5, 65	StdForwarding	83
Debugging	8	string	79, 86
Konfigurationsdatei	6	StringList	86
ProxyPort	6	subFolders	83, 84
ProxyTimeout	6	Suchbegriff	99
receivers	72	Suche	66
redirectionDescription	87	Syntaxbeschreibung	66, 67, 69, 98
reference	79	System.Web.Services	13
register	77, 78	System.Xml	13
RegisterDescription	85	table	67
replace	77, 78	task	72, 85, 87
report	66	tasks	61
report2	67	Tasks	98
report3	69	testSession	74
Request	9	text	79
reserved	84	timeAbsolute	47
Response	9	timeAbsoluteSpecified	47
responseDate	85	Timeout	65
returnStatus	71	Title	98
returnValue	71	Trefferliste	67, 69
rows	84	triggerAction	73
runScript	71	trustedLogin	74
Schema		Typbezeichner	56
key	95	type	79
Search_DateFrom	99	typeOption	64
Search_DateUntil	99	Überschreiben	43, 45
Search_Fulltext	99	ungleich	99

Unterordner	29	Web Services Enhancements	13
Url	5, 11	Webservice	5, 92
Ursprungarchives	86	Wiedervorlage	
Ursprungsarchiv	76, 82	Datum setzen	47
user	64, 74, 87	Wiedervorlageordner	47, 98
userInfo	75	wishedFieldNames	52, 53
utf-8	15	Workflow	83, 86
valid	74	beenden	30
value	78	starten	73
values	84	WorkflowAction	86
Vergleichsoperatoren	99	workflowActions	63
Verknüpfungen	99	WorkflowIdent	86
Versendung	98	workflowinfo	39, 82
Versendungshistorie	73	describeFileType2	41
version	86	FileTypeDescription2	82
versioning	78	WorkflowPattern	62, 86
View		Workflowschritt	63, 81
Key	95	WorkflowStep	86
Visual Basic	10	WorkflowSteps	63
Visual Studio 2005®	10	WSDL	9, 10, 87
Visual Studio 2008®	10	wsdl.exe	10
Volltextsuche	99	Zielarchivs	81
Von	99	Zuletzt benutzt	98
wasRedirected	87		

7. Abbildungsverzeichnis

Abb. 1: Kommandozeilenaufruf docsoaproxy -h.....	6
Abb. 2: docsoaproxy.ini als Konfigurationsdatei.....	6
Abb. 3: Sequenzdiagramm eines Proxy-Timeouts.....	7
Abb. 4: Wechsel der SessionIds im Rahmen eines Login- Logouts	13
Abb. 5: Auswahl des Startobjektes für das Projekt	14
Abb. 6: Einstellen der Unicode - utf-8 Codepage 65001	15
Abb. 7: Einstellen der Konsolenschriftart- Lucida Console	16
Abb. 8: Erstellen einer C# Konsolenanwendung	22
Abb. 9: Hinzufügen eines Dienstverweises	22
Abb. 10: Veranschaulichung eines Service References	23
Abb. 11: EE.i Archiv-Key mit Detaildialog bei importierter Archivstruktur	93
Abb. 12: EE.x View-Key mit Detaildialog bei importierter Archivstruktur	94
Abb. 13: EE.x Schema-Key mit Detaildialog bei importierter Archivstruktur.....	95
Abb. 14: Archivserver EAS	95
Abb. 15: Sequenzdiagramm eines Client-Timeouts	100