

# Content Composer

## Web Services

### Foundation 22.1

Written by: Documentation Team, R&D

Date: Tuesday, June 21, 2022

## Documentation Notice

Information in this document is subject to change without notice. The software described in this document is furnished only under a separate license agreement and may only be used or copied according to the terms of such agreement. It is against the law to copy the software except as specifically allowed in the license agreement. This document or accompanying materials may contain certain information which is confidential information of Hyland Software, Inc. and its affiliates, and which may be subject to the confidentiality provisions agreed to by you.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright law, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Hyland Software, Inc. or one of its affiliates.

Hyland, HXP, OnBase, Alfresco, Nuxeo, and product names are registered and/or unregistered trademarks of Hyland Software, Inc. and its affiliates in the United States and other countries. All other trademarks, service marks, trade names and products of other companies are the property of their respective owners.

© 2022 Hyland Software, Inc. and its affiliates.

The information in this document may contain technology as defined by the Export Administration Regulations (EAR) and could be subject to the Export Control Laws of the U.S. Government including for the EAR and trade and economic sanctions maintained by the Office of Foreign Assets Control as well as the export controls laws of your entity's local jurisdiction. Transfer of such technology by any means to a foreign person, whether in the United States or abroad, could require export licensing or other approval from the U.S. Government and the export authority of your entity's jurisdiction. You are responsible for ensuring that you have any required approvals prior to export.

# Inhalt

Informationen zu den Content Composer Web Services .....	6
Informationen zur Architektur .....	6
WS Process .....	6
<i>Informationen zum Process Identifier</i> .....	6
<i>Optionen des Vorgangs</i> .....	7
<i>Informationen zur Adresse eines Objekts im Bündel</i> .....	9
<i>Aktionen, Kommandos und deren Ergebnisse</i> .....	9
Aktionen der Web Services bei Verwendung des MWS_Standard-Prozesses .....	9
Dynamische Abhängigkeiten .....	10
Vorgangsteuerung .....	11
Fehlerbehandlung .....	11
Kommandos und deren Ergebnisse .....	11
<i>Informationen zum Vorgang</i> .....	14
<i>Informationen zur Vorgangweiterleitung</i> .....	19
<i>Die Action-Eigenschaft state</i> .....	20
<i>Die Action-Eigenschaft configured</i> .....	20
<i>Spezielle Attribute der Aktion PRINTDOCUMENTS</i> .....	20
<i>Spezielle Attribute der Aktion DATENBESCHAFFUNG</i> .....	21
<i>Poolvariablen abfragen und setzen</i> .....	22
XML Beispiele .....	23
Eingebettete Aktionen im Object XML .....	23
Beispiele für die Verwendung von Aktionen .....	24
Error XML .....	26
Info XML .....	27
Login XML .....	29
MwsDataProviderDefinition XML .....	29
MwsFolderContent XML .....	30
MwsItemInfo XML .....	30
MwsLookupResult XML .....	30
MwsNavigList XML .....	31
MwsNavigTree Options XML .....	32

MwsNavigTree XML .....	32
MwsServerInfo XML .....	32
MwsSystemList XML .....	33
MwsValueHelpDefinition XML .....	33
Object XML .....	33
Optionen beim Schließen eines Vorgangs .....	37
Optionen beim Öffnen eines Vorgangs .....	37
Optionen beim Schließen eines Vorgangs .....	37
Processlist Options XML .....	37
Erweiterte, optionale XML Attribute .....	38
Processlist XML .....	41
Systemdef XML .....	42
UserInfo XML .....	42
<i>Funktionen</i> .....	45
ClearCache .....	45
Doc_GetFile_Mime .....	46
Doc_SetFile_Mime .....	47
Login .....	47
Logout .....	48
Obj_GetStructure .....	49
Obj_SetStructure .....	50
Obj_Toggle .....	51
Ping .....	51
Process_Close .....	52
Process_Create .....	53
Process_Delete .....	54
Process_Forward .....	55
Process_GetInfo .....	55
Process_GetLastError .....	56
Process_GetList .....	57
Process_Open .....	58

Process_SetData .....	59
Process_Start .....	59
Rep_CreateItem .....	61
Rep_CreateVersion .....	62
Rep_GetBinFile_Mime .....	63
Rep_GetDataProviderDefinition .....	64
Rep_GetFolderContent .....	65
Rep_GetForms .....	66
Rep_GetItemDescription .....	66
Rep_GetLookupObjects .....	68
Rep_GetLookupValue .....	68
Rep_GetNavigList .....	69
Rep_GetNavigTree .....	70
Rep_GetPrinters .....	71
Rep_GetSystem .....	72
Rep_GetSystemList .....	72
Rep_GetValueHelpDefinition .....	73
Rep_LockItem .....	73
Rep_ReadItemBlob .....	74
Rep_UnLockItem .....	75
Rep_UpdateItemBlob .....	76
Srv_GetInfo .....	77
Usr_GetRoles .....	78
Usr_GetUsers .....	78
Usr_GetUsersOfRole .....	79
WS UserRepository .....	80
Rückgabewerte von MWS Funktionen .....	80
<i>Ergebnis bei erfolgreicher Ausführung</i> .....	80
<i>Fehlercodes</i> .....	80
<i>Warnungen</i> .....	82
Informationen zur Integration .....	82
Informationen zum Repository .....	83
Korrespondenzvorgänge .....	83

## Informationen zu den Content Composer Web Services

Die Content Composer Web Services (MWS) setzen eine lauffähige Content Composer Studio Installation voraus.

MWS stellt für Drittanwendungen über den Standard SOAP (Simple Object Access Protocol) verschiedene Funktionen eines Korrespondenzsystems bereit.

Die Web Services ermöglichen es Drittanwendungen, folgende Aufgaben auszuführen:

- Sitzungen serverseitig zu öffnen und zu schließen
- Informationen über ein Repository zu beziehen.

Weitere Informationen finden Sie unter [Repository](#)

- Korrespondenzvorgänge zu initiieren und zu steuern

Weitere Informationen finden Sie unter [Correspondence Processes](#).

Auf der Basis der Web-Services werden sowohl klassische Webanwendungen als auch Thin Clients (Windows-Anwendungen die direkt auf das SOAP Interface zugreifen) entwickelt.

Weitere Informationen finden Sie unter [Integration](#).

## Informationen zur Architektur

Die Web Services lassen sich aus technischer Sicht in drei Schichten (Layer) unterteilen:.

- Der Interface Layer übernimmt die Veröffentlichung der Web Services und regelt den Zugriff auf diese. Die Kommunikation mit der Außenwelt erfolgt ausschließlich über SOAP
- Im Business Layer liegt die Geschäftslogik der Korrespondenzerstellung. Hier erfolgt der Zugriff auf das Repository und die Dokumenterstellung. Auch die Kommunikation mit dem Daten Server übernimmt der Business Layer. Wie im Content Composer Studio ist es möglich, über Xdata definierte Datenselektionen aufzurufen oder XML-Daten zu importieren.
- Der State Layer ist für die Speicherung von Repository-, Status- und Prozessdaten zuständig und ist somit die einheitliche Basis für die Textadministration und die Sachbearbeiterfunktionalität.

## WS Process

### Informationen zum Process Identifier

Der Process Identifier (PID) ist eine eindeutige Nummer eines Vorganges. Unter dieser Nummer werden alle Vorgangsdaten auf dem Server gehalten. Diese muss daher systemweit eindeutig sein. Der PID darf maximal 40 Zeichen lang und muss als Dateiname verwendbar sein.

Ein Vorgang muss immer mit der Funktion `Process_Delete` gelöscht werden, damit die Ressourcen auf dem Server freigegeben werden. Nach dem Aufruf von `Process_Delete` ist der PID nicht mehr verwendbar.

## Optionen des Vorgangs

Mit den Optionen, die immer als XML-Widestring übergeben werden müssen, können diverse Einstellungen am Vorgang vorgenommen werden. Sie müssen immer als XML-Widestrings übergeben werden. Der Aufbau des XML Strings ähnelt dem XML Result der Funktion `Process_GetInfo` und spiegelt die interne Objektstruktur der MWS wieder. Weitere Informationen finden Sie unter [Info XML](#).

Beim Setzen der Optionen dürfen nur die Attribute gesetzt werden, die in diesem Abschnitt explizit als Attribute gekennzeichnet werden. Da es sich um einen XML-Widestring handelt, können viele unterschiedliche Einstellungen gleichzeitig getroffen werden.

### Verfügbare Einstellungen

- Setzen des Startkommandos

Weitere Informationen finden Sie unter [Process\\_Create](#), Element `mwsprocessmgr`, Attribut `onstart`.

Mögliche Werte des Attributs: Aktionen, Kommandos und deren Ergebnisse

- Setzen des Titels und der Beschreibung

Weitere Informationen finden Sie unter [Process\\_Close](#) und [Optionen beim Schließen eines Vorgangs](#).

- Setzen von allgemeinen Einstellungen der Aktionen

- Jede Aktion eines Vorgangs ist konfigurierbar.
- Es gibt Aktionen, die eine Konfiguration erfordern und Aktionen, die nicht zwingend konfiguriert werden müssen.
- Bei allen Aktionen (Element `mwsaction`) können die Attribute `state` und `configured` gesetzt werden.
- Durch das Setzen der Eigenschaft `state` einer Aktion von `finished` auf den Wert `ready` kann eine Aktion wiederholt werden.

**Hinweis:** Bereits vorhandene Ergebnisse (z.B. Dokumente) dieser und aller nachfolgenden Aktionen werden gelöscht.

- Durch das Setzen der Eigenschaft `configured` einer Aktion auf den Wert `true`, kann festgelegt werden, dass die Konfiguration dieser Aktion abgeschlossen ist.
- Setzen von Parametern für die Datenbeschaffung

- Die Parameter für die Datenbeschaffung gehören zur Aktion **DATASELECTION** und müssen deshalb als Unterelement dieser Aktion angegeben werden.
- Das Mengenelement für alle Parameter der Datenbeschaffung hat den Namen **selparams**.
- Die Parameter werden über das Attribut **name** adressiert. Der Wert muss als C\_DATA übergeben werden.

#### Beispiel

```
...
<selparams>
  <selparam name="Partner-Nr">0815</selparam>
</selparams>
...
```

- Setzen von manuellen Variablen
  - Die manuellen Variablen gehören zur Aktion **CREATEDOCUMENTS** und müssen deshalb als Unterelement dieser Aktion angegeben werden.
  - Das Mengenelement für alle manuellen Variablen des Bündels hat den Namen **manvars**.
  - Das Mengenelement für alle manuellen Variablen eines Dokuments hat den Namen **docref**.
  - Das Dokument ist mit dem Attribut **reference** zu adressieren. Weitere Informationen finden Sie unter [Adresse eines Objekts im Bündel](#).
  - Die manuelle Variable selbst wird im Element mit dem Namen **manvar** beschrieben und ist ebenfalls durch das Attribut **reference** zu adressieren.

Der Wert muss als C\_DATA übergeben werden.

#### Beispiel

```
...
<manvars>
  <docref reference="790B817A-B2C3-475E-86E5-08118150EA94">
    <manvar reference="DocumentCollection.790B817A-B2C3-475E-86E5-08118150EA94:308DFBC5-F520-414E-ACBD-0A4495C33301_E4437CCD-E71A-4181-9C6E-0691B26E0C8FValue1/manvar"
    <manvar reference="DocumentCollection.790B817A-B2C3-475E-86E5-08118150EA94:308DFBC5-F520-414E-ACBD-0A4495C33301_EAAC39D9-81BA-476B-A003-039ABF3618AAValue2/manvar"
    </docref>
  </manvars>
...
```

- Ausgabe von Parametern für das Outputmanagement.

Parameter für das Outputmanagement gehören zur Aktion **PRINTANDARCHIVE** und müssen deshalb als Attribute dieser Aktion angegeben werden.

Weitere Informationen finden Sie unter **Spezielle Attribute der Aktion PRINTDOCUMENTS**

- Setzen von Variablen im Pool SYSTEMPOOL

Poolvariablen sind unabhängig von Aktionen und können daher direkt unter dem Element **process** angegeben werden.

Weitere Informationen finden Sie unter [Get and Set Pool Variables](#).

Folgende Attribute dürfen generell nicht gesetzt werden:

- process\_status
- lastaction
- lastactionname
- lasterrorcode
- lasterrormessage

## Informationen zur Adresse eines Objekts im Bündel

Jedes Objekt in einem Bündel besitzt eine eindeutigen Referenz. Diese Referenz wird im Attribut reference angegeben. Die jeweilige Referenz setzt sich aus den ChildIds der Parent-Objekte und des jeweiligen Objektes zusammen.

### Beispiel

```
790B817A-B2C3-475E-86E5-08118150EA94:308DFBC5-F520-414E-ACBD-0A4495C33301_
EAAC39D9-81BA-476B-A003-039ABF3618AA
```

## Aktionen, Kommandos und deren Ergebnisse

Ein Vorgang der Web Services wird im Wesentlichen durch seine Aktionen definiert. Eine Aktion kann Vorgänger haben und kann eine Konfiguration erfordern. Eine Aktion hat außerdem einen Status, der Informationen über den Arbeitsfortschritt der Aktion erteilt.

Innerhalb eines MWS-Prozesses repräsentieren die vorhandenen MwsActivityStatus-Prozesskomponenten die Aktionen.

### Aktionen der Web Services bei Verwendung des MWS\_Standard-Prozesses

Aktion	Beschreibung
OBJECTSELECTION	Objektauswahl
DATASELECTION	Datenbeschaffung

Aktion	Beschreibung
COMPLETESTRUCTURE	Bündelstruktur ergänzen z.B. dynamisches Bündel
CREATEDOCUMENTS	Dokumenterstellung
FORWARD	Vorgang weiterleiten
PRINTDOCUMENTS	Bündeldokumente drucken
PRINTANDARCHIVE	Übergabe der Bündeldokumente an Odin

Diese Reihenfolge gibt auch die Abhängigkeit der Aktionen untereinander vor:

- Es kann erst gedruckt werden, wenn die Dokumente erstellt worden sind.
- Die Dokumente können erst erstellt werden, wenn bekannt ist, welche Dokumente erstellt werden sollen.
- Erst wenn die Daten beschafft worden sind, ist bekannt welche Dokumente erstellt werden sollen.

Zusätzlich zu dieser festen Abhängigkeit zwischen den Aktionen gibt es auch noch dynamische Abhängigkeiten, die nicht immer vorhanden sind und oftmals nur durch eine Benutzereingabe gelöst werden können.

Dies bedeutet, dass eine Aktion in bestimmten Fällen erst dann ausgeführt werden kann, wenn eine Konfiguration dieser Aktion durchgeführt worden ist.

### Dynamische Abhängigkeiten

- Daten können erst dann beschafft werden, wenn die Parameter für die Selektion bekannt sind.  
Die Notwendigkeit zur Eingabe dieser Parameter ist aber erst dann gegeben, wenn solche Parameter überhaupt vorhanden sind.
- Dokumente können erst dann erstellt werden, wenn die manuellen Variablen gesetzt worden sind.  
Die Notwendigkeit zur Eingabe von manuellen Variablen ist aber erst dann gegeben, wenn manuelle Variablen vorhanden sind. Dies kann sich bekanntlich, je nachdem welche Textbausteine ausgewählt worden sind, sehr schnell ändern.
- Ein Bündel kann nur dann sofort gedruckt werden, wenn der Sofortdrucker angegeben wurde.  
Die Notwendigkeit zur Angabe eines Sofortdruckers ist erst dann gegeben, wenn Sofortdruck gewünscht ist.
- Ein Bündel kann nur dann weitergeleitet werden, wenn entweder ein Benutzer und/oder eine Benutzergruppe angegeben ist, an die der Vorgang weitergeleitet werden soll.  
Ist eine oder mehrere dieser dynamischen Abhängigkeiten gegeben, so kann die jeweilige Aktion erst dann erfolgreich gestartet werden, wenn die Konfiguration abgeschlossen wurde.

Eine Aktion gilt erst dann als konfiguriert, wenn die Eigenschaft (siehe Action-Eigenschaft configured) entsprechend gesetzt wurde.

Eigenschaften von Aktionen können in den Optionen (siehe Optionen des Vorgangs) beim Aufruf der Funktionen [Process\\_Start](#) oder [Process\\_Create](#) übermittelt werden.

## Vorgangsteuerung

Die Vorgangsteuerung der Web Services sieht vor, dass jede Aktion prinzipiell immer gestartet werden kann. Wurden die Vorgänger der zu startenden Aktion noch nicht ausgeführt, dann versucht die Vorgangsteuerung diese auszuführen. Die Überprüfung der Vorgängeraktionen startet immer bei der ersten Aktion des Prozesses.

Wenn also beispielsweise direkt nach der Erstellung des Vorgangs **PRINTDOCUMENTS** gestartet wird, dann versucht die Vorgangsteuerung alle noch ausstehenden Aktionen, beginnend bei der ersten Aktion des Prozesses, auszuführen.

Sollte eine dieser Aktionen nicht konfiguriert sein, wird eine entsprechende Configuration-Warning zurückgegeben. Ansonsten wird die Aktion ausgeführt und die nächste Aktion wird geprüft.

Der Vorgang läuft somit bis zur ersten Configuration-Warning oder endet mit der Ausführung der gestarteten Ziel-Aktion **PRINTDOCUMENTS**.

## Fehlerbehandlung

Sind manuelle Eingaben erforderlich, führt der Start eines Kommandos zur Rückgabe des Fehlercodes 6000.

Damit nach der Eingabe der fehlenden Informationen das ursprünglich gestartete Kommando fortgesetzt werden kann, merken sich die MWS dieses. Beim nächsten Kommandostart kann das gespeicherte Kommando durch das fiktive Kommando `CONTINUE_LAST_ACTION` erneut aktiviert werden.

## Beispiel Pseudocode

```
...
res := Process.Process_Start(FSID, FPID, 'PRINT_AND_ARCHIVE', options,
resxml);
if res = 6000 then
begin
    ok := LetUserConfigActions(options, resxml);
    if ok then
        res := Process.Process_Start(FSID, FPID, 'CONTINUE_LAST_ACTION',
options, resxml);
end;
...
```

## Kommandos und deren Ergebnisse

Wie man am letzten Beispiel sehen kann, entspricht der Inhalt des Parameters **command** nicht dem Namen der Aktionen des Vorgangs.

Abgesehen von den zwei statischen Kommandos **SET\_PARAMETERS** und **CONTINUE\_LAST\_ACTION** leiten sich die Kommandos aus den Namen der im MWS-Prozess enthaltenen Skriptausführungs-ProzessBausteine ab.

Dabei wird durch Angabe des Kommandos der erste Skriptausführungs-ProzessBaustein, dessen Objektnamen mit dem übergebenen Kommando übereinstimmt, als Zielaktivität festgelegt.

Wird kein entsprechender Skriptausführungs-ProzessBaustein gefunden, wird ein entsprechender Fehler ausgelöst.

**Die folgende Auflistung bezieht sich auf den MWS\_Standard Prozess.**

- **SET\_PARAMETERS**

Bei diesem Kommando werden nur die Parameter in den Optionen ausgewertet und gesetzt. Eine Aktion wird nicht gestartet. Bei Erfolg wird ein leerer String zurückgegeben.

- **SELECT\_OBJECT**

Die Aktion **OBJECTSELECTION** wird gestartet. Bei Erfolg gleicht das Ergebnis XML dem des Kommandos **SELECT\_DATA**.

Ist kein Bündel geladen und entsprechende Parameter zum Laden des Objektes (Bündel) benötigt werden, kehrt das Kommando mit dem Fehlercode 6000 zurück.

Die Beschreibung zum Aufbau des XMLs finden Sie im Kapitel XML Beispiele einer Configuration Warning für die Aktion **FORWARD**.

- **SELECT\_DATA**

Die Aktion **DATASELECTION** (Datenbeschaffung) wird gestartet. Bei Erfolg wird ein XML zurückgegeben, welches die Struktur des Bündels widerspiegelt.

Dieses XML entspricht dem Ergebnis der Prozedur **Obj\_GetStructure** wenn man im Parameter **RootRef** einen Leerstring und im Parameter **MaxLevel** eine 1 übergeben würde.

Das Kommando kehrt mit dem Fehlercode 6000 zurück, wenn Parameter für den Start der Selektion benötigt werden.

In diesem Fall enthält das Ergebnis-XML die benötigten Parameter.

Die Beschreibung zum Aufbau des XMLs finden Sie im Kapitel XML Beispiele zu **SELPARAMS**.

- **COMPLETE\_STRUCTURE**

Das Ereignis **NachDatenbeschaffung** wird ausgelöst und das entsprechende Skript (sofern zugewiesen) wird ausgeführt.

Das Ergebnis-XML entspricht dem des Kommandos **SELECT\_DATA**.

- **CREATE\_DOCUMENTS**

Die Dokumenterstellung wird gestartet. Auch bei diesem Kommando wird, wie beim Kommando **SELECT\_DATA**, bei Erfolg die Struktur des Bündels zurückgegeben.

Das Kommando kehrt mit dem Fehlercode 6000 zurück, wenn noch manuelle Variablen benötigt werden.

In diesem Fall enthält das Ergebnis-XML die benötigten manuellen Variablen.

Die Beschreibung zum Aufbau des XMLs finden Sie im Kapitel XML Beispiele zu MANVARS.

- FORWARD\_DOCUMENTS

Die Aktion **FORWARD** wird ausgeführt. Bei Erfolg gleicht das Ergebnis XML dem des Kommandos **SELECT\_DATA**.

Das Kommando kehrt mit dem Fehlercode 6000 zurück, wenn das Bündel geblockt ist (im Skript **NachDokumenteErzeugen** wurde das Flag **BlockBundle** auf true gesetzt und der aktuelle User entspricht dem User, der den Vorgang angelegt hat).

In diesem Fall enthält das Ergebnis-XML die benötigten Parameter zur Vorgangweiterleitung.

Die Beschreibung zum Aufbau des XMLs finden Sie im Kapitel XML Beispiele zu SELPARAMS.

- PRINT\_DOCUMENTS

Führt die Aktion PRINTDOCUMENTS aus. Die Übergabe in das Outputmanagement wird vorbereitet.

Bei Erfolg gleicht das Ergebnis XML dem des Kommandos **SELECT\_DATA**.

Das Kommando kehrt mit dem Fehlercode 6000 zurück, wenn nicht für alle Dokumente gültige Druckdefinitionen vorhanden sind.

In diesem Fall enthält das Ergebnis-XML die Dokumente deren Druckdefinition noch mittels der Methode **Obj\_SetStructure** zu vervollständigen sind.

Genauere Informationen zum Aufbau des XMLs finden Sie im Kapitel OUTPUTPARAMS.

- PRINT\_AND\_ARCHIVE

Die Übergabe in das Outputmanagement wird mittels der **OdinProcessImport**-Komponente durchgeführt.

Bei Erfolg gleicht das Ergebnis XML dem des Kommandos **SELECT\_DATA**.

Anderenfalls wird ein entsprechendes Fehler-XML zurückgegeben.

- CONTINUE\_LAST\_ACTION

Wenn ein Kommando durch den Fehler 6000 abgebrochen wurde, dann kann es mit Hilfe dieses fiktiven Kommandos wieder gestartet werden.

Liegt kein gespeichertes Kommando vor, wird ein entsprechendes Error XML zurückgegeben.

## Informationen zum Vorgang

Ein Vorgang in den Web Services besteht aus mehreren Aktionen, aus einem Vorgangsmanager, der die Aktionen verwaltet, sowie aus Variablenpools. Jedes dieser Objekte hat Eigenschaften, die sich im Laufe des Vorgangs verändern können.

Eine Anwendung die die MWS integriert, hat die Möglichkeit Informationen zum Vorgang abzurufen, um sie anzuzeigen bzw. um zu entscheiden, welche weiteren Aktionen auszuführen sind. Damit die Datenmenge die über die Leitung gesendet wird, minimal gehalten werden kann, besteht die Möglichkeit, die Informationen auch selektiv anzufordern. Dazu muss der Typ der benötigten Information angegeben werden.

Die Eigenschaften des Vorgangsmanagers werden unabhängig vom Informationstyp immer zurückgegeben.

Das Element `mwsprocessmngnr` befindet sich unter dem Element `process`.

### Beispiel

```
...
<mwsprocessmngnr lasterrorcode="0" lasterrormessage="" lastactionname=""
onstart="">
...
```

Attribut	Beschreibung
lasterrorcode	Nummer des zuletzt aufgetretenen Fehlers
lasterrormessage	Beschreibung des zuletzt aufgetretenen Fehlers
onstart	Name der Aktion, die direkt nach dem Erstellen des Vorgangs gestartet werden soll. <b>Hinweis:</b> Das Setzen dieser Eigenschaft zu einem späteren Zeitpunkt als beim Erstellen des Vorgangs hat keine Auswirkung.
lastactionname	Name der zuletzt ausgeführten Aktion

Eine weitere wichtige Information ist der zuletzt aufgetretene Fehler. Diese Information wird immer dann unabhängig vom Informationstyp zurückgegeben, wenn ein Fehler aufgetreten ist. Diese Information befindet sich unmittelbar unter dem Element `mws`.

```
...
<error errorcode="6000">error text</error>
...
```

Aktuell werden folgende Informationstypen unterstützt

### PROCESSINFO

Attribut	Beschreibung
Titel	Titel des Vorgangs
Beschreibung	Beschreibung des Vorgangs

Unmittelbar unter dem Vorgangsmanger werden mit dem Element **mwsaction** die Aktionen aufgelistet.

Jede Aktion hat folgende Attribute:

Attribut	Beschreibung
name	Name der Aktion
lasterrorcode	Nummer des zuletzt aufgetretenen Fehlers
lasterrormessage	Beschreibung des zuletzt aufgetretenen Fehlers
state	Siehe <a href="#">Action Eigenschaft state</a>
configured	Siehe <a href="#">Action Eigenschaft configured</a>

Erweiterte Attribute der Aktion **DATASELECTION** Siehe [Spezielle Attribute der Aktion DATASELECTION](#)

Erweiterte Attribute der Aktion **PRINTANDARCHIVE** Siehe [Spezielle Attribute der Aktion PRINTDOCUMENTS](#)

## POOLVARS

Ausgabe von Variablen des Pools **SYSTEMPOOL**. Siehe [Poolvariablen abfragen und setzen](#).

## MANVARS, MANVARS2

Ausgabe von manuellen Variablen des Bündels.

Da die manuellen Variablen zur Aktion **CREATEDOCUMENTS** gehören, werden sie unter dem entsprechenden Element aufgelistet.

## Beispiel

```
...
<mwsaction name="CREATEDOCUMENTS" lasterrorcode="0" lasterrormessage=""
state="0" configured="0">
  <manvars>
    <docref name="Partner_Application_ManVar" title="Partner_Application_
ManVar" reference="790B817A-B2C3-475E-86E5-08118150EA94">
      <manvar name="Agent_Forename"
reference="DocumentCollection.790B817A-B2C3-475E-86E5-08118150EA94:308DFBC5-
```

```

F520-414E-ACBD-0A4495C33301_E4437CCD-E71A-4181-9C6E-0691B26E0C8F"
title="Agent_Forename" mask="" vh_name="" vh_id="" vh_system="" vh_type="2"
input_enabled="Y" data_link="MODUSUSER" data_attribute="Forename" input_
required="0"/>
    <manvar name="Agent_Surname" reference="DocumentCollection.790B817A-
B2C3-475E-86E5-08118150EA94:308DFBC5-F520-414E-ACBD-0A4495C33301_EAAC39D9-
81BA-476B-A003-039ABF3618AA" title="Agent_Surname" mask="" vh_name="" vh_
id="" vh_system="" vh_type="0" input_enabled="Y" data_link="MODUSUSER" data_
attribute="Surname" input_required="0"/>Doe
    </docref>
</manvars>
<manvars2>
    <docref name="Partner_Application_ManVar" title="Partner_Application_
ManVar" reference="790B817A-B2C3-475E-86E5-08118150EA94">
        <manvar2 name="Agent_Forename"
reference="DocumentCollection.790B817A-B2C3-475E-86E5-08118150EA94:308DFBC5-
F520-414E-ACBD-0A4495C33301_E4437CCD-E71A-4181-9C6E-0691B26E0C8F"
title="Agent_Forename" mask="" vh_name="Forename" vh_id="" vh_system="" vh_
type="2" input_enabled="Y" data_link="MODUSUSER" data_attribute="Forename"
input_required="0"/><valuehelp>&lt;combobox
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
sorted="true">
    <items>
        <item>Martin</item>
        <item>Hugo</item>
        <item>Ernest</item>
        <item>Max</item>
        <item>John</item>
    </items>
</combobox></valuehelp>Doe</manvar2>
        <manvar2 name="Agent_Surname"
reference="DocumentCollection.790B817A-B2C3-475E-86E5-08118150EA94:308DFBC5-
F520-414E-ACBD-0A4495C33301_EAAC39D9-81BA-476B-A003-039ABF3618AA"
title="Agent_Surname" mask="" vh_name="" vh_id="" vh_system="" vh_type="0"
input_enabled="Y" data_link="MODUSUSER" data_attribute="Surname" input_
required="0"/>
            </docref>
        </manvars2>
</mwsaction>
...

```

Die Elemente **manvars** und **manvars2** sind Sammelknoten für alle manuellen Variablen. Die manuellen Variablen eines Dokuments werden unter dem Sammelknoten **docref** aufgelistet.

Die Elemente **manvars** und **manvars2** haben folgende Attribute:

Attribut	Beschreibung
name	Name des Dokuments
title	Titel des Dokuments.
reference	Siehe <a href="#">Adresse eines Objekts im Bündel</a>

Die Elemente **manvars** und **manvars2** beschreiben eine manuelle Variable wie folgt:

Attribut	Beschreibung
name	Name der manuellen Variablen
reference	Siehe <a href="#">Adresse eines Objekts im Bündel</a>
title	Titel für den Prompt
mask	Eingabemaske (siehe Studio Dokumentation)
vh_name	Name der Werthilfe (optional)
vh_id	Objekt-ID der Werthilfe (optional)
vh_system	System-ID der Werthilfe (optional)
vh_type	Typ der Werthilfe <b>Mögliche Werte</b> <b>0:</b> Der Variablen ist keine Werthilfe zugewiesen <b>1:</b> Textbox <b>2:</b> Combobox <b>3:</b> Kalender <b>4:</b> Numerisches Eingabefeld (SpinEdit) <b>7:</b> Mehrzeilige Textbox
input_enabled	0, wenn keine manuelle Eingabe möglich sein soll (wenn z.B. nur eine Auswahl möglich sein soll) Dieses Attribut ist nur dann vorhanden, wenn keine manuelle Eingabe möglich sein soll.
Data	Aktueller Wert der manuellen Variablen

Falls in den zugehörigen AppSettings angegeben, muss das Element `manvar2` vom MWS bei Anfrage von `manual variables info` zurückgegeben werden. Dies ist der Fall für

- Die zurückgegebene Warnung `Configuration for action CREATEDOCUMENTS required!`
- Einen Aufruf von `Process_GetInfo` mit dem `infoType` **MANVARS**.

Ist eine Werthilfe der Variablen zugewiesen, die durch das Element `manvar2` beschrieben wird, wird ein zusätzliches Kindelement `<valuehelp>` hinzugefügt, das die zugehörige serialisierte Werthilfedefinition enthält.

## SELPARAMS

Ausgabe von Parametern für die Datenbeschaffung. Da diese Parameter zur Aktion `DATASELECTION` gehören, werden sie unter dem entsprechenden Element aufgelistet.

### Beispiel

```
...
<mwsaction name="DATASELECTION" lasterrorcode="0" lasterrormessage=""
state="0" configured="0">
  <selparams>
    <selections>
      <selection name="Partner_Letter" title="Partner_Letter"
active="True">
        <selparam name="Partnernumber" displayname="PartnerNumberTest"
mask="" />0815</selparam>
      </selection>
    </selections>
  </selparams>
</mwsaction>
```

Ein Parameter für die Datenbeschaffung wird mit dem Element `selparam` beschrieben:

Attribut	Beschreibung
<code>name</code>	Name des Parameters
<code>displayname</code>	Name, der in der Anzeige verwendet werden soll. Wenn leer, wird der Name verwendet.
<code>mask</code>	Eingabemaske für diesen Parameter. Wenn leer, wird keine Eingabemaske verwendet.
<code>data</code>	Parameterwert

## OUTPARAMS

Ausgabe von Parametern für das Outputmanagement. Da diese Parameter zur Aktion `PRINTDOCUMENTS` gehören, werden sie unter dem entsprechenden Element aufgelistet.

### Beispiel

```

...
<mwsaction name="PRINTDOCUMENTS" lasterrorcode="0" lasterrormessage=""
state="0" configured="0">
  <objects>
    <object title="ApplicationPack" type="D" name="ApplicationPack"
index="07102009-289-tdr-dm1" reference="933110B7-0485-4C62-A50C-
0B9D7FC62DB1"/>
  </objects>
</mwsaction>
...

```

Es sind alle Objekte aufgelistet, deren Druckdefinition noch mit Hilfe der Methode **Obj\_SetStructure** zu vervollständigen sind.

## Informationen zur Vorgangswweiterleitung

Dieses XML wird von der Funktion **Process\_Start** zurückgegeben, wenn das **PrintAndArchive** Kommando auf ein geblocktes Bündel angewendet wird. Ein Bündel ist dann geblockt, wenn im Skript **NachDokumenteErzeugen** das Flag **BlockBundle** auf true gesetzt wurde und der aktuelle Benutzer derjenige ist, der den Vorgang erstellt hat.

Um dem Client die Durchführung einer automatischen Weiterleitung zu ermöglichen, kann im Skript **NachDokumenteErzeugen** ein Weiterleitungsziel angegeben werden (Variable DestUser und/oder DestUsergroup).

Die dort gesetzten Weiterleitungsziele werden in den entsprechenden Attributen zurückgegeben, so dass sie als Parameter zum Aufruf der Funktion **Process\_Forward** verwendet werden können. Es wird mindestens ein Weiterleitungsziel benötigt. Wurden keine entsprechenden Weiterleitungsziele im Skript **NachDokumenteErzeugen** definiert, muss der Client eine entsprechende Auswahl implementieren.

### Beispiel

```

<mws type="FORWARD" version="2">
  <error errorcode="6000">Configuration for action FORWARD required!</error>
  <process id="120de98f-9ec4-469a-b7ad-376651986aa9" systemoid="dm1">
    <title>BasicBindingTest</title>
    <mwsprocessmgr lasterrorcode="6000" lasterrormessage="Configuration
for action FORWARD required!" lastactionname="PRINT_AND_ARCHIVE" onstart="">
      <mwsaction name="FORWARD" lasterrorcode="0" lasterrormessage=""
state="0" configured="0"><forwardparams destuser="MyDestUser"
destusergroup="MyDestGroup"/></mwsaction>
    </mwsprocessmgr>
  </process>
</mws>

```

Attribut	Beschreibung
destuser	Der im Skript <b>NachDokumenteErzeugen</b> angegebene Benutzer, an den der Vorgang weitergeleitet werden soll.
destusergroup	Die im Skript <b>NachDokumenteErzeugen</b> angegebene Benutzergruppe, an die der Vorgang weitergeleitet werden soll.

### Die Action-Eigenschaft *state*

Die folgende Tabelle listet die möglichen Werte der Action-eigenschaft *state* auf.

Wert	Beschreibung
0	unknown (es liegen noch keine Statusinformationen vor)
1	ready (bereit zur Ausführung)
2	busy (befindet sich in der Ausführung)
3	finished (Ausführung ist abgeschlossen)

### Die Action-Eigenschaft *configured*

Die folgende Tabelle listet die möglichen Werte der Action-eigenschaft *configured* auf.

Wert	Beschreibung
0	false (nicht konfiguriert)
1	true (konfiguriert)

### Spezielle Attribute der Aktion PRINTDOCUMENTS

Mit der Methode `Obj_SetStructure` besteht die Möglichkeit, die Druckdefinition eines Dokumentes zu setzen bzw. zu ändern.

Attribut	Wert	Beschreibung
print_asktime	0	Abfrage des Druckzeitpunktes soll erfolgen.
	1	Abfrage des Druckzeitpunktes soll nicht erfolgen.

Attribut	Wert	Beschreibung
form		Formularname
printer		Druckername
printer_type	OEP OP	Email-Drucker Online-Drucker
printtime	0 1 2	Unbekannt Sofort Versetzt
print_changeable	0 1	Änderungen der Druckdefinition erlaubt. Änderungen der Druckdefinition nicht erlaubt.

**Beispiel**

```
<object reference="790B817A-B2C3-475E-86E5-08118150EA94"
printer="OnlinePrinter" printer_type="OP" form="Form1" copycount="0"
copytext="" printtime="2" />
```

**Spezielle Attribute der Aktion DATENBESCHAFFUNG**

Mit der Methode Obj\_SetStructure besteht die Möglichkeit, die Druckdefinition eines Dokumentes zu setzen bzw. zu ändern.

Attribut	Beschreibung
selection	Name der Selektion

**Beispiel**

```
<selections>
  <selection name="Partner_Letter" title="Partner_Letter" active="True">
    <selparam name="Partnernumber" displayname="PartnerNumberTest"
mask="" />0815</selparam>
  </selection>
  <selection name="Partner_V2" title="Partner_V2" active="False"/>
</selections>
```

Element	Beschreibung
selections	Sammelement aller vorhanden Datenreferenzen/Selektionen im Bündel.
selection	Beschreibung einer Datenreferenz mit den Attributen. name: Name der Datenreferenz title: Titel der Datenreferenz active: true: Aktive Selektionen false: XML-Selektion (Daten werden mittels der Methode SetData an den MWS übergeben)
selparams	Parameterbeschreibung bei aktiver Selektion mit den Attributen: name: Name des Parameters displayname: Angezeigter Name des Parameters mask: Eingabemaske DATA: aktueller Wert des Parameters, wenn bereits gesetzt.

## Poolvariablen abfragen und setzen

Poolvariablen können im Rahmen einer Informationsbeschaffung mittels der Methode Process\_GetInfo (infoType POOLVARS) abgefragt werden.

Dabei werden der SYSTEMPOOL- und der MODUSUSER-Pool zurückgegeben.

Gesetzt werden können nur Variablen des Pools SYSTEMPOOL.

Poolvariablen dieses Pools werden mit der Methode Process\_Start gesetzt.

Wenn nur die Poolvariablen gesetzt werden sollen und keine weitere Aktion ausgeführt werden soll, dann muss das Kommando SET\_PARAMETERS verwendet werden.

### Beispiel XML

```
<pools>
  <pool name="SYSTEMPOOL"/>
    <poolvar name="MyVariable" type="S">MyVariableValue</poolvar>
  </pool>
</pools>
...
```

Element	Beschreibung
pools	Sammelknoten für alle Pools

Element	Beschreibung
pool	Sammelknoten für alle Poolvariablen eines bestimmten Pools. Das Attribut <b>name</b> enthält den Namen des Pools.
poolvar	Beschreibung einer Poolvariablen mit den Attributen: <b>name</b> : Name der Poolvariablen <b>type</b> : B: Boolean (TRUE oder FALSE) L: Longinteger (Zahlen) S: Widestring (beliebige Zeichenkette) F: Float (Gleitkommazahlen, Dezimaltrennzeichen je nach Ländereinstellung) <b>data</b> : Inhalt der Poolvariablen

**Hinweis**

Beim Setzen von Variablen ist das Attribut **type** optional.

Wird das Attribut nicht angegeben, dann wird der Typ **Widestring** angenommen.

Wird ein Typ angegeben, muss der im Data-Bereich angegebene Inhalt zum Datentyp passen.

**XML Beispiele****Eingebettete Aktionen im Object XML**

Werden mit Set\_Structure bestimmte Objekteigenschaften gesetzt, können auch Aktionen auf das Bündel oder Dokument durchgeführt werden.

**Hinweis:** Für dynamisch eingefügte Dokumente und Bausteine entspricht der automatisch beim Einfügen vergebene Referenzstring einer negativen Zahl.

**Attributbeschreibung**

Attribut	Beschreibung
Aktion	<b>includeDoc</b> : Neues Dokument dem Bündel hinzufügen. (Sachbearbeiter Dokument) <b>includeBlock</b> : Dynamisches Baustein- Einfügen in Dokument des Bündels <b>move</b> : Vorhandenes Objekt (Sachbearbeiter-Dokumente und Bausteine) verschieben. <b>delete</b> : Eingefügtes Objekt (Sachbearbeiter-Dokumente und Bausteine) löschen.

Attribut	Beschreibung
	<b>loadexternaldocument:</b> Laden / entfernen des externen Dokuments
target_ reference	Referenzindex des Objektes im Baum, der die Einfügeposition bestimmt. Wird dieser Wert nicht angegeben, wird das Objekt am Ende eingefügt.
position	<b>after:</b> Objekt wird nach der target_reference eingefügt. <b>before:</b> Objekt wird vor der target_reference eingefügt.
name	Name des Objektes das eingefügt werden soll, bzw. bei loadexternaldocument eindeutiger Name der Datei, die bereits zuvor hochgeladen wurde oder Leerstring, um das externe Dokument wieder zu lösen.
reference	Für die Aktionen Move/Delete die Referenz des Objektes das verschoben/gelöscht werden soll. Für die Aktion loadexternaldocument die Referenz des Objektes, bei dem das externe Dokument zugeordnet/gelöst werden soll.
doc_ reference	Referenz des Dokumentes, in das ein Baustein dynamisch eingefügt werden soll.

### Hinweise

- Stellen Sie clientseitig sicher, dass die bei der Action loadexternaldocument mit dem Attribut name angegebene Datei zuvor vom Client mit der Methode Doc\_SetFile\_Mime auf den Server hochgeladen wurde.
- Der Client muss einen eindeutigen Namen für die Datei setzen, um Dateiüberschreibungen durch Namensgleichheit zu vermeiden.
- Soll eine externe Datei wieder gelöst werden, so muss bei der Aktion loadexternaldocument ein Leerstring als Wert des Attributs name verwendet werden.

### Beispiele für die Verwendung von Aktionen

#### Einfügen eines Dokuments

```
<mws type="OBJECT">
  <process>
    <object action="includedoc" name="TestDocument" target_reference=""
position="after"/>
  </process>
</mws>
```

#### Verschieben eines Dokuments

```
<mws type="OBJECT">
  <process>
    <object action="move" reference="-3" target_reference="-1"
position="before"/>
  </process>
</mws>
```

### **Löschen eines Dokuments**

```
<mws type="OBJECT">
  <process>
    <object action="delete" reference="-1"/>
  </process>
</mws>
```

### **Einfügen eines Bausteins (ohne Zielobjekt)**

```
<mws type="OBJECT">
  <process>
    <object action="includeblock" name="DynBst1" doc_reference="-1" target_
reference="-1" position="unknown" />
  </process>
</mws>
```

### **Einfügen eines Bausteins (vor Zielobjekt)**

```
<mws type="OBJECT">
  <process>
    <object action="includeblock" name="DynBlock1" doc_reference="-1"
target_reference="-1" position="unknown" />
  </process>
</mws>
```

### **Verschieben eines Bausteins**

```
<mws type="OBJECT">
  <process>
    <object action="move" reference="-1:-2" target_reference="-1:-1"
position="after"/>
  </process>
</mws>
```

### **Löschen eines Bausteins**

```
<mws type="OBJECT">
  <process>
    <object action="delete" reference="-1:-2"/>
  </process>
</mws>
```

### **Setzen eines externen Dokuments für einen Baustein**

Die angegebene Datei muss zuvor vom Client zum Server hochgeladen worden sein (Methode: Doc\_SetFile\_Mime).

```
<mws type="OBJECT">
  <process>
    <object action="loadexternaldocument"
name="MyExternalBlockDocument.docx" reference="9BD927E4-39CC-4217-90D1-
38C3B15F4CED:35E5A161-D291-4244-84FF-6CB30F14D1A0" />
  </process>
</mws>
```

### Lösen eines externen Dokuments bei einem Baustein

```
<mws type="OBJECT">
  <process>
    <object action="loadexternaldocument" name="" reference="9BD927E4-39CC-
4217-90D1-38C3B15F4CED:35E5A161-D291-4244-84FF-6CB30F14D1A0" />
  </process>
</mws>
```

### Lösen eines externen Dokuments bei einem Dokument

Die angegebene Datei muss zuvor vom Client zum Server hochgeladen worden sein (Methode: Doc\_SetFile\_Mime).

```
<mws type="OBJECT">
  <process>
    <object action="loadexternaldocument"
name="MyExternalStaticDocument.pdf" reference="C6D5FD76-516C-482E-AAA6-
9E51BE8BD2E2" />
  </process>
</mws>
```

### Lösen eines externen Dokuments bei einem Dokument

```
<mws type="OBJECT">
  <process>
    <object action="loadexternaldocument" name="" reference="C6D5FD76-516C-
482E-AAA6-9E51BE8BD2E2" />
  </process>
</mws>
```

### Error XML

```
<?xml version="1.0" encoding="utf-16"?>
<mws xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" type="ERROR" version="2.0">
  <error location="functionname" errorcode="numeric Code">error
message</error>
</mws>
```

## Info XML

```

<mws type="PROCESSINFO,SELPARAMS,MANVARS,POOLVARS" version="2">
  <process id="a0deab2e-7c05-498e-8c92-0648cff1b3da" systemoid="dm1">
    <title>BasicBindingTest</title>
    <mwsprocessmngtr lasterrorcode="0" lasterrormessage="" lastactionname=""
onstart="">
      <mwsaction name="FORWARD" lasterrorcode="0" lasterrormessage=""
state="0" configured="0"/>
      <mwsaction name="PRINTANDARCHIVE" lasterrorcode="0"
lasterrormessage="" state="0" configured="0"/>
      <mwsaction name="PRINTDOCUMENTS" lasterrorcode="0"
lasterrormessage="" state="0" configured="0"/>
      <mwsaction name="OBJECTSELECTION" lasterrorcode="0"
lasterrormessage="" state="0" configured="0"/>
      <mwsaction name="DATASELECTION" lasterrorcode="0"
lasterrormessage="" state="0" configured="0"/>
      <selparams>
        <selections>
          <selection name="Partner_Letter" title="Partner_Letter"
active="True">
            <selparam name="Partnernumber"
displayname="PartnerNumberTest" mask=""/>
          </selection>
          <selection name="Partner_V2" title="Partner_V2"
active="False"/>
        </selections>
      </selparams>
    </mwsaction>
    <mwsaction name="COMPLETESTRUCTURE" lasterrorcode="0"
lasterrormessage="" state="0" configured="0"/>
    <mwsaction name="CREATEDOCUMENTS" lasterrorcode="0"
lasterrormessage="" state="0" configured="0">
      <manvars>
        <docref name="Partner_Application_ManVar" title="Partner_
Application_ManVar" reference="790B817A-B2C3-475E-86E5-08118150EA94">
          <manvar name="Agent_Forename"
reference="DocumentCollection.790B817A-B2C3-475E-86E5-08118150EA94:308DFBC5-
F520-414E-ACBD-0A4495C33301_E4437CCD-E71A-4181-9C6E-0691B26E0C8F"
title="Agent_Forename" mask="" vh_name="" vh_id="" vh_system="" vh_type="2"
input_enabled="Y" data_link="MODUSUSER" data_attribute="Forename" input_
required="0"/>
          <manvar name="Agent_Surname"
reference="DocumentCollection.790B817A-B2C3-475E-86E5-08118150EA94:308DFBC5-
F520-414E-ACBD-0A4495C33301_EAAC39D9-81BA-476B-A003-039ABF3618AA"
title="Agent_Surname" mask="" vh_name="" vh_id="" vh_system="" vh_type="0"

```

```

input_enabled="Y" data_link="MODUSUSER" data_attribute="Surname" input_
required="0"/>
    </docref>
    <docref name="Partner_Application" title="Partner_Application"
reference="933110B7-0485-4C62-A50C-0B9D7FC62DB1"/>
    </manvars>
    <manvars2>
        <docref name="Partner_Application_ManVar" title="Partner_
Application_ManVar" reference="790B817A-B2C3-475E-86E5-08118150EA94">
            <manvar2 name="Agent_Forename"
reference="DocumentCollection.790B817A-B2C3-475E-86E5-08118150EA94:308DFBC5-
F520-414E-ACBD-0A4495C33301_E4437CCD-E71A-4181-9C6E-0691B26E0C8F"
title="Agent_Forename" mask="" vh_name="" vh_id="" vh_system="" vh_type="2"
input_enabled="Y" data_link="MODUSUSER" data_attribute="Forename" input_
required="0"/><valuehelp>
<combobox xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
sorted="true">
    <items>
        <item>Martin</item>
        <item>Hugo</item>
        <item>Ernest</item>
        <item>Max</item>
    </items>
</combobox></valuehelp></manvar2>
            <manvar2 name="Agent_Surname"
reference="DocumentCollection.790B817A-B2C3-475E-86E5-08118150EA94:308DFBC5-
F520-414E-ACBD-0A4495C33301_EAAC39D9-81BA-476B-A003-039ABF3618AA"
title="Agent_Surname" mask="" vh_name="" vh_id="" vh_system="" vh_type="0"
input_enabled="Y" data_link="MODUSUSER" data_attribute="Surname" input_
required="0"/>
            </docref>
            <docref name="Partner_Application" title="Partner_Application"
reference="933110B7-0485-4C62-A50C-0B9D7FC62DB1"/>
        </manvars2>
    </mwsaction>
</mwsprocessmgr>
<pools>
    <pool name="SYSTEMPOOL"/>
    <pool name="MODUSUSER">
        <poolvar name="USERID" type="S">MMueller</poolvar>
        <poolvar name="USERROLENAMES" type="S">Administrator_Role,User_
Role</poolvar>
        <poolvar name="EMAIL" type="S">n.v. mail</poolvar>
        <poolvar name="TELEPHONE" type="S">n.v. telephonenumber</poolvar>
        <poolvar name="USERPRINCIPALNAME" type="S">n.v.

```

```

userprincipalname</poolvar>
    <poolvar name="DISPLAYNAME" type="S">n.v. DisplayName</poolvar>
DisplayName</poolvar>
    </pool>
</pools>
</process>
</mws>

```

## Login XML

Die [Login](#) Funktion übergibt den Benutzernamen und das Passwort zur Authentifizierung.

In dieser **Login XML** können Sie einen Alias-Benutzernamen angeben, den die Sitzung beim Erstellen einer neuen Kommunikation als Benutzer zuweist.

```

<?xml version="1.0" encoding="utf-16"?>
<mws>
    <loginparams>
        <loginparam name=""alias">USERNAME</loginparam>
    </loginparams>
</mws>

```

## MwsDataProviderDefinition XML

```

mws xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" type="DATAPROVIDERDEF"
version="2.0">
    <providerxml>&lt;?xml version="1.0" encoding="utf-
16"?&gt;&lt;ms:dataproviderdefinitiontree xmlns:ms="http://www.composer-
suite.de/2006/XMLSchema"&gt;&lt;dataproviderdefinition objectid="09072009-28-
dpd-mws" objectname="All_SQL_MWS_Sessions" description="" xml="" category=""
rowsperpage="50" tables="MWS_SESSION" dbalias="MWS|21012008-1-dba-
mws|2|DbAlias|DatabaseAlias|mws" protected="False"&gt;&lt;sqlcolumns
objectname="SqlColumnCollection" description=""&gt;&lt;sqlcolumn
objectname="MWSS_SESSION_ID" description="" title="" dbcolumnname="MWSS_
SESSION_ID" size="-1" isvisible="True" /&gt;&lt;sqlcolumn objectname="MWSS_
CREATED" description="" title="" dbcolumnname="MWSS_CREATED" size="-1"
isvisible="True" /&gt;&lt;sqlcolumn objectname="MWSS_USERNAME" description=""
title="" dbcolumnname="MWSS_USERNAME" size="-1" isvisible="True"
/&gt;&lt;sqlcolumn objectname="MWSS_COMPOSERUSERINFO" description="" title=""
dbcolumnname="MWSS_COMPOSERUSERINFO" size="-1" isvisible="True"
/&gt;&lt;/sqlcolumns&gt;&lt;/dataproviderdefinition&gt;&lt;/ms:dataproviderde
finitiontree&gt;
    </providerxml>
</mws>

```

## MwsFolderContent XML

```
<mws xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" type="FOLDERCONTENT"
version="2.0">
<foldercontent systemname="it2">
  <items>
    <item name="Test1_1" objectid="19102009-1230-f-it2" type="A"/>
    <item name="TextBlock" objectid="it2-tc-17" type="B"/>
    <item name="TextBlockWithVariable" objectid="it2-tc-18" type="B"/>
  </items>
</foldercontent>
</mws>
```

## MwsItemInfo XML

```
<mws xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" type="ITEMINFO" version="2.0">
  <iteminfo>
    <VersionNrString>1.1</VersionNrString>
    <VersionNr>10001</VersionNr>
    <VersionLabel>This is the VersionLabel</VersionLabel>
    <IsReleaseVersion>false</IsReleaseVersion>
    <RevisionNumber>2893</RevisionNumber>
    <IsProtected>false</IsProtected>
    <IsReadOnly>false</IsReadOnly>
    <ObjectName>Partner_Antragsannahme</ObjectName>
    <ObjectId>02062008-805-td-it11</ObjectId>
    <DatabaseId>294</DatabaseId>
    <ObjectType>TextDocument</ObjectType>
    <LastModifiedAt>2012-09-25T11:10:23.62</LastModifiedAt>
    <LastModifiedBy>TestUser</LastModifiedBy>
    <CreatedAt>2012-03-07T13:16:59.5</CreatedAt>
    <CreatedBy>TestUser</CreatedBy>
    <LockedBy />
    <LockedAt xsi:nil="true" />
    <Context>None</Context>
  </iteminfo>
</mws>
```

## MwsLookupResult XML

**Bei Rep\_GetLookupValue (beinhaltet MwsLookupValue xml):**

```
<mws xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" type="LOOKUPRES" version="2.0"
reference="VariablenVorgabe|15042008-666-1t-it2|447|LookupTable|None|dm1">
  <lookup aliastable="DefaultVariables" alias="Var1">Wert1|"\"(999 99\
99999;1;_</lookup>
</mws>
```

### Bei Rep\_GetLookUpObjects (beinhaltet MwsFolderContent xml):

```
<mws xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" type="LOOKUPRES" version="2.0"
reference="SystemAliasTable|15042008-665-at-it2|446|AliasTable|None|dm1">
  <foldercontent systemname="dm1">
    <items>
      <item name="DefaulVariables" objectid="15042008-666-1t-it2"
type=""/>
    <items>
  </foldercontent>
</mws>
```

### MwsNavigList XML

Dieses XML wird beim Navigieren durch das Repository verwendet.

```
<mws xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" type="NAVIGLIST" version="2.0">
  <foldercontent path="" filter="A,P,D,B" systemname="itl2">
    <items>
      <item name="Paragraph1" objectid="14102009-1768-tc-itl2" type="B" />
      <item name="Paragraph2" objectid="14102009-1769-tc-itl2" type="B" />
      <item name="KeepParagraphsTogether" objectid="14102009-1767-tc-itl2"
type="B" />
      <item name="KeepParagraphsTogetherTables" "14102009-1770-tc-itl2"
type="B" />
      <item name="Sender_Date" objectid="14102009-1759-tc-itl2" type="B"
/>
      <item name="Partner_Application" objectid="14102009-1790-td-itl2"
type="D" />
      <item name="Partner_Application_XDOC" objectid="14102009-1799-td-
itl2" type="D" />
      <item name="XML-Testdocument" objectid="14102009-1798-td-itl2"
type="D" />
      <item name="Application" objectid="14102009-1807-tdc-itl2" type="P"
/>
      <item name="Application_XDOC" objectid="14102009-1812-tdc-itl2"
type="P" />
    <items>
  </mws>
```

## MwsNavigTree Options XML

```
<mws>
  <navigtree>
    <item name="startfoldername" value="[OrdnerName]" />
  </navigtree>
</mws>
```

Durch die Angabe eines Ordnernamens in wird angegeben, von welchem Ordner ausgehend der Verzeichnisbaum / die Ordner-Hierarchie geliefert werden soll.

Wird ein Leerstring übergeben, wird die Angabe ignoriert: d.h. es wird die komplette Hierarchie ausgehend vom RootFolder des Systems zurückgeliefert.

Existiert der angegebene Ordner in dem System nicht, wird ein entsprechendes ErrorXml zurückgegeben.

## MwsNavigTree XML

```
<mws xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" type="NAVIGTREE"
version="2.0" systemname="it2">
  <folder name="ImportTest_2" type="A" path="/" dbid="0">
    <folders>
      <folder name="Test1" type="A" path="/Test1/" dbid="126">
        <folders>
          <folder name="Test1_1" type="A" path="/Test1/Test1_1/"
dbid="2371"/>
        </folders>
      </folder>
      <folder name="Test2" type="A" path="/Test2/" dbid="897"/>
    </folders>
  </folder>
</mws>
```

## MwsServerInfo XML

```
<mws xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
type="SERVERINFO" version="2.0">
  <serverinfo>
    <expiredate>2110-12-31</expiredate>
    <licensefeatures>
      <feature>Basic</feature>
      <feature>Xdata</feature>
      <feature>Odin</feature>
      <feature>ObjectVersioning</feature>
    </licensefeatures>
  </serverinfo>
</mws>
```

## MwsSystemList XML

```
<mws xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" type="SYSTEMLIST" version="2.0">
  <systems>
    <system name="MWS_Standard" description="" systemoid="mws"/>
    <system name="Batch_Standard" description="" systemoid="modb"/>
    <system name="User_Standard" description="" systemoid="user"/>
    <system name="Odin_Standard" description="" systemoid="odv"/>
    <system name="TestSystem" description="" systemoid="ts"/>
    <system name="Test" description="" systemoid="test"/>
  </systems>
</mws>
```

## MwsValueHelpDefinition XML

```
<mws type="VALUEHELP" version="2.0" valuehelptype="3" reference="Date_
Time|27102008-182-vhl-dm|15|ValueHelp|None|odv">
  <definition>&lt;datetimepicker
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd=http://www.w3.org/2001/XMLSchema customformat="dd.MM.yyyy HH:mm:ss"
format="Custom" /&gt;
  </definition>
</mws>
```

## Object XML

```
<?xml version="1.0" encoding="utf-16"?>
<mws type="OBJECT">
  <process id="ba56e16a-b7ba-495f-8ab9-158e48df5846">
    <object title="COD6773_test_DIPWrite" type="P" name="COD6773_test_
DIPWrite" index="12112018-20207-tdc-obatest" reference="">
      <object title="COD6773_Simple" type="D" name="COD6773_Simple"
index="12112018-20208-tdr-obatest" reference="ED90176C-B7FF-46F8-A1B4-
0D00F508B36C" state="1" enabled="1" print_asktime="0" form="COD6773_Html_
Body" printer="COD6773" printer_type="OEP" printtime="1" print_changeable="1"
copycount="0" copy="0" password="" doctype="1" dataid="Partner_V2" external_
doctype_id="2" external_doctype_name="doc" filename="COD6773_Simple_ED90176C-
B7FF-46F8-A1B4-0D00F508B36C_O.docx" isprotected="0"/>
      <object title="COD6773_Simple_2" type="D" name="COD6773_Simple_2"
index="12112018-20209-tdr-obatest" reference="19CA3FBB-1BFC-4122-B26D-
67A8F503BF83" state="2" enabled="1" print_asktime="0" form="COD6773_Html_
Body" printer="COD6773" printer_type="OEP" printtime="1" print_changeable="1"
copycount="0" copy="0" password="" doctype="1" dataid="Partner_V2" external_
doctype_id="2" external_doctype_name="doc" filename="COD6773_Simple_2_
19CA3FBB-1BFC-4122-B26D-67A8F503BF83_O.docx" isprotected="0"/>
      <object title="COD6773_Simple" type="D" name="COD6773_Simple"
```

```

index="12112018-20210-tcr-obatest" reference="C4C8FDDB-2D27-4956-A6EA-
BA2F08904433" state="1" enabled="1" print_asktime="0" form="COD6773_Html_
Body" printer="COD6773" printer_type="OEP" printtime="1" print_changeable="1"
copycount="0" copy="1" password="" doctype="1" external_doctype_id="23"
external_doctype_name="doc" copytext="" originalid="ED90176C-B7FF-46F8-A1B4-
0D00F508B36C"/>
  <object title="COD6773_Simple_3" type="D" name="COD6773_Simple_3"
index="20012020-20414-tdr-obatest" reference="7781B1FE-2F66-4B9D-9F7B-
997E9E97424C" state="2" enabled="1" print_asktime="0" form="COD6773_Html_
Body" printer="COD6773" printer_type="OEP" printtime="1" print_changeable="0"
copycount="0" copy="0" password="" doctype="1" dataid="Partner_V2"
filename="COD6773_Simple_3_7781B1FE-2F66-4B9D-9F7B-997E9E97424C_O.docx"
isprotected="0"/>
  </object>
</process>
</mws>

```

### Beschreibung der gemeinsamen Attribute des Elements object

Element	Beschreibung
title	Titel (Ersatz für den Namen zur Anzeige für den Anwender)
type	Objekttyp P = Bündel D = Dokumentvorlage B = Baustein
name	Objektname
index	Objektindex
reference	Referenz des Dokuments im Bündel Siehe: <a href="#">Adresse eines Objekts im Bündel</a>

### Zusätzliche Attribute bei Dokumenten, Kopien und Bausteinen

Element	Beschreibung
state	<b>0:</b> Deactivated (Objekt ist deaktiviert, kann aber aktiviert werden) <b>1:</b> Required (Objekt ist aktiviert und darf nicht deaktiviert werden) <b>2:</b> Activated (Objekt ist aktiviert, kann aber deaktiviert werden)

Element	Beschreibung
	<b>3:</b> Disabled (Objekt ist deaktiviert und darf nicht aktiviert werden)
enabled	<p><b>0</b></p> <ul style="list-style-type: none"> <li>• beim Dokument: wenn state gleich deactivated oder disabled ist</li> <li>• bei der Dokument-Kopie: wenn das dazugehörige Original nicht aktiviert ist (Original hat den state deactivated oder disabled)</li> </ul> <p><b>1</b></p> <ul style="list-style-type: none"> <li>• beim Dokument: wenn das Dokument aktiviert ist (state ist activated oder required)</li> <li>• bei der Kopie: wenn Original aktiviert ist und Kopie den state activated oder required hat</li> </ul>

#### Zusätzliche Attribute bei Dokumenten / Kopien

Element	Beschreibung
print_asktime	<p><b>0:</b> Abfrage des Druckzeitpunktes soll erfolgen.</p> <p><b>1:</b> Abfrage des Druckzeitpunktes soll nicht erfolgen.</p>
form	Formularname
printer	Druckername
printer_type	<p><b>OEP:</b> Email-Drucker</p> <p><b>OP:</b> Online-Drucker</p>
printtime	<p><b>0:</b> Unbekannt</p> <p><b>1:</b> Sofort</p> <p><b>2:</b> Versetzt</p>
print_changeable	<p><b>0:</b> Änderungen der Druckdefinition erlaubt.</p> <p><b>1:</b> Änderungen der Druckdefinition nicht erlaubt.</p>
copyCount	Anzahl echter Kopien
copy	<p><b>0:</b> Original</p> <p><b>1:</b> Kopie</p>

Element	Beschreibung
password	Kennwort des Dokumentes im Klartext
doctype	<b>1:</b> DynamicWord <b>2:</b> StaticWord <b>3:</b> StaticPdf <b>4:</b> StaticTiff <b>5:</b> StaticXps
external_doctype_id	ID des OnBase Dokumenttyps (long)
external_doctype_name	Name des OnBase Dokumenttyps (string)

**Das folgende Attribut ist nur bei Originalen enthalten**

Element	Beschreibung
dataid	Name der zugeordneten TextDatenReferenz, die die zu verwendende Selektion referenziert.

**Die folgenden Attribute sind nur bei Kopien enthalten**

Element	Beschreibung
copytext	Text auf Kopie
originalid	ChildId des Originaldokumentes. Der Wert korrespondiert mit dem Wert des Attributes reference des entsprechenden Originaldokumentes.

**Folgende Attribute sind nur enthalten, wenn das Dokument bzw. die Kopie bereits erzeugt wurden**

Element	Beschreibung
filename	Name des erzeugten Dokumentes
isprotected	<b>0:</b> Der Dokumentenschutz ist nicht aktiviert <b>1:</b> Der Dokumentenschutz ist aktiviert

**Zusätzliche Attribute bei Bausteinen**

Element	Beschreibung
childcount	Anzahl der Kinder des Bausteins

**Folgendes Attribut ist bei Dokumenten / Bausteinen nur enthalten, wenn dem Objekt mittels des Clients ein externes Dokument zugeordnet wurde**

Element	Beschreibung
isexternaldocument	1: Es ist ein externes Dokument zugeordnet.

### Optionen beim Schließen eines Vorgangs

```
<mws version="2">
  <title>Title of the process</title>
  <description>Description of the process</description>
</mws>
```

### Optionen beim Öffnen eines Vorgangs

```
<mws>
  <openoptions>
    <option name="checkstate" value="true" />
  </openoptions>
</mws>
```

### Optionen beim Schließen eines Vorgangs

### Processlist Options XML

Mit diesem Options-XML kann angegeben werden, welche Vorgänge in der zurückgegebenen Vorgangsliste enthalten sein sollen.

#### Mögliche Einstellung

1. Rückgabe der vom Benutzer selbst erzeugten Vorgänge (default bei Übergabe eines Leerstrings).
2. Rückgabe der Vorgänge die an den Benutzer direkt oder an eine Benutzergruppe, der der Benutzer angehört, weitergeleitet wurden.
3. Rückgabe der eigenen sowie der weitergeleiteten Vorgänge. Wird für diesen Parameter ein Leerstring übergeben, so werden nur die vom Benutzer selbst erzeugten Vorgänge (siehe Punkt 1) zurückgegeben.

```

<mws>
  <processlist>
    <item name="forwarded" include="1" />
    <item name="nonforwarded" include="0"/>
  </processlist>
</mws>

```

Die Items des Elementes processlist legen fest, welche Vorgänge in der zurückgegebenen Vorgangsliste enthalten sind.

#### Die Attribute haben folgende Bedeutung:

Element	Beschreibung
name	<p><b>forwarded:</b> Es werden alle weitergeleiteten Vorgänge zurückgegeben. Dies sind die Vorgänge, die entweder an den Benutzer selbst oder an eine Benutzergruppe, der der Benutzer angehört, weitergeleitet wurden.</p> <p><b>nonforwarded:</b> Es werden alle nicht weitergeleiteten Vorgänge zurückgegeben. Dies sind die Vorgänge, die der Benutzer selbst erstellt hat.</p>
include	<p><b>0:</b> Die entsprechenden Vorgänge werden nicht in die Rückgabeliste aufgenommen.</p> <p><b>1:</b> Die durch das Attribut name spezifizierten Vorgänge sollen in der Liste enthalten sein.</p>

#### Besonderheiten

Wird für die Options ein Leerstring übergeben, oder werden beide Attribute ausgeschaltet (include="0"), dann werden defaultmäßig nur die vom Benutzer selbst erstellten Vorgänge zurückgegeben.

#### Erweiterte, optionale XML Attribute

Optional können die folgenden Attribute zur weiteren Anpassung der zurückgelieferten Vorgangsliste angegeben werden:

```

<mws>
  <processlist>
    ...
    <item name="listuser" include="Admin" />
    <item name="shortprocdesc" include="10"/>
    <item name="additionalCondition" include=" AND (MWS_STATUS = 2) " />
    <item name="replacecondition" include="(MWS_STATUS = 2) " />
    ...
  </processlist>
</mws>

```

#### Die Attribute haben folgende Bedeutung:

Element	Beschreibung
name	<p><b>listuser:</b></p> <p>Wird dieses Attribut spezifiziert, wird die entsprechende Prozessliste des angegebenen Users zurückgegeben. Dabei werden bei den weitergeleiteten Prozessen nur diejenigen berücksichtigt, die an den angegebenen User direkt weitergeleitet wurden.</p> <p><b>shortprocdesc:</b></p> <p>Wird dieses Attribut spezifiziert und ein Prozess hat eine Prozessbeschreibung, ist in seinem process-Element innerhalb des zurückgegebenen ProcessList XML das Attribut shortprocdesc enthalten.</p> <p>Das Attribut shortprocdesc ist mit den ersten x Zeichen der Prozessbeschreibung befüllt.</p> <p>Ist keine Prozessbeschreibung vorhanden, ist das entsprechende Attribut nicht enthalten.</p> <p><b>Beispiel</b></p> <pre data-bbox="386 827 1179 1052">&lt;process id="ade9e10d-ccfb-490f-a5b4-6fb1ac69bfd8"   user="Admin" system="dml" type=""   objectname="TestBundle" title="My Title"   state="0" created_at="27012010140555"   lastsaved="04022010173422"   created_by="Admin"   shortprocdesc="ShortDescription" /&gt;</pre> <p><b>additionalcondition:</b></p> <p>Wird dieses Attribut spezifiziert, wird die Bedingung des SelectStatements um die angegebene Bedingung erweitert.</p> <p>Damit können die Bedingungen, die sich aufgrund anderer processlist options ergeben, individuell erweitert werden.</p> <p><b>replacecondition:</b></p> <p>Wird dieses Attribut spezifiziert, wird die Bedingung des SelectStatements komplett durch die hier angegebene Bedingung ersetzt.</p> <p>Alle anderen spezifizierten processlist options werden überschrieben.</p> <p>Dadurch besteht die Möglichkeit, innerhalb der Prozessliste genau die Prozesse zu bekommen, die die angegebene Bedingung erfüllen.</p>
include	<p><b>Username:</b> Enthält dieses Attribut einen Leerstring, wird es ignoriert</p> <p><b>Anzahl Zeichen:</b> Anzahl der zurückzugebenden Zeichen der Prozessbeschreibung. Enthält dieses Attribut den Wert 0, wird es ignoriert.</p> <p><b>Hinzuzufügende Bedingung:</b> Enthält dieses Attribut einen Leerstring, wird es ignoriert</p> <p><b>Neue Bedingung:</b> Enthält dieses Attribut einen Leerstring, wird es ignoriert</p>

## Anwendungsbeispiel

In diesem Beispielskript werden alle Prozesse gelöscht, die entweder abgeschlossen (Status = 2) oder gesperrt (Status=1) sind und länger als einen Tag nicht mehr bearbeitet wurden.

Dazu wird mittels der Prozesslist Option `replacecondition` die Bedingung für das `Select-Statement` komplett neu definiert und anschließend alle in der erhaltenen Prozessliste vorhandenen Prozesse gelöscht.

```
;***** MLMwsClient *****
GetObject("MwsClient","MLMwsClient")
MwsClient.MwsUrl = "http://localhost:8011/mws/mwsprocess"
GetObject("date", "MLDate")
;
; Current date
CurrDateAsInt = date.CurrentDate()
;
; current date minus 1 day
SearchDateInt = date.IncDate(CurrDateAsInt, -1, 0, 0)
;
;SearchDate as string
SearchDateStr = date.Int64ToDateString("dd.MM.yyyy", SearchDateInt)
FreeObject("date")
Protocol(SearchDateStr, 8)
;
; replacecondition --> Status = 1 and last edit less than SearchDate
options = FormatStr("<mws><processlist><item name=""replacecondition""
include=""(MWS_STATUS = 2) OR ((MWS_STATUS = 1) AND (MWS_LASTSAVED &lt;=
'%s'))"" /></processlist></mws>", SearchDateStr)
myXmlResult = ""
res = MwsClient.Process_GetList(0, 100, options)
;
Protocol(res, 8)
myXmlResult = MwsClient.LastXmlResult
Protocol(result, 8)
;
if (myXmlResult <> "")
  GetObject("xml", "MLXmlDocument")
  xml.LoadXml(myXmlResult)
  root = xml.DocumentElement ;rootElement holen
  nodeList = root.SelectNodes("/mws/processlist/process") ;get all nodes of
the process list
  Protocol("Number of processes: {0}", 8, nodeList.Count)
  nodeListEnumerator = nodeList.GetEnumerator() ; Get an enumerator for the
nodes
  ;
  ; Iterate over all items...
  readNext = nodeListEnumerator.MoveNext()
```

```

while (readNext)
  node = nodeListEnumerator.Current ; get current item node
  ;
  ;read processid
  attrColl = node.Attributes
  attr = attrColl.GetNamedItem("id")
  processId = attr.Value
  Protocol("ProcessId: {0}", 8, processId)
  if (processId <> "")
    MwsClient.Process_Delete(processId) ; Delete process
    Protocol("Process [{0}] deleted", 8, processId)
  end-if
  readNext = nodeListEnumerator.MoveNext() ; get next item node
end-while
FreeObject("xml")
end-if
FreeObject("MwsClient")
;

```

## Processlist XML

```

<mws xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" type="PROCESSLIST"
version="2.0" eof="true" from="1" to="100">
  <processlist>
    <process id="0eba3667-a70a-4a65-bf02-74dd3be7e44a" user="User1"
system="demo" type="" objectname="Request" objectid="12022008-1362-tdc-
moddemo" title="" state="0" created_at="16122019102153"
lastsaved="16122019104544" created_by="User1" destinationuser="Administrator"
destinationgroup="AdminRole" />
    <process id="f8f63cae-7246-4f43-8be8-9f4edd5ff91c" user="User2"
system="demo" type="" objectname="Request" objectid="12022008-1362-tdc-
moddemo" title="fssf" state="0" created_at="27112019130125"
lastsaved="27112019130136" created_by="User2" destinationgroup="AdminRole" />
    <process id="3d4a7c31-f710-4b24-9006-2896715a9eba" user="User2"
system="obatest" type="" objectname="COD6773_test_DIPWrite"
objectid="12112018-20207-tdc-obatest" title="WebApi Forward" state="0"
created_at="15042019182725" lastsaved="07052019174335" created_by="User2"
destinationuser="BuildAdmin" />
    <process id="9a6bd514-25fe-431d-b1f9-bacf94128a68" user="BuildAdmin"
system="demo" type="" objectname="Request" objectid="12022008-1362-tdc-
moddemo" title="" state="1" created_at="18112019113712"
lastsaved="18112019113712" created_by="BuildAdmin" />
  </processlist>
</mws>

```

### Attributes of the element 'process'

Element	Beschreibung
id	Vorgangs ID
user	Benutzername
system	SystemOld des geladenen Bündels
type	Vergebener Vorgangstyp. Dieses Attribut ist nur vorhanden, wenn ein Typ angegeben wurde.
objectname	Name des geladenen Objekts.
title	Vergebener Vorgangstitel. Dieses Attribut ist nur vorhanden, wenn ein Titel angegeben wurde.
state	Status des Vorgangs
created_at	Datum der Vorgangserzeugung. Datums-Format: DDMMYYYYHHMMNN
lastsaved	Datum der letzten Benutzeraktivität an diesem Vorgang. Datums-Format: DDMMYYYYHHMMNN
created_by	Name des Benutzers, der den Vorgang erstellt hat.
destinationuser	Benutzer, an den der Vorgang weitergeleitet wurde. Dieses Attribut ist nur vorhanden, wenn der Vorgang an einen Benutzer weitergeleitet wurde..
destinationgroup	Gruppe, an die der Vorgang weitergeleitet wurde. Dieses Attribut ist nur vorhanden, wenn der Vorgang an eine Gruppe weitergeleitet wurde..

## Systemdef XML

```
<?xml version="1.0" encoding="UTF-8"?>
<mws xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" type="SYSTEMDEF" version="2.0">
  <systemdef name="Odin_Standard" description="" systemoid="odv"/>
</mws>
```

## Userinfo XML

```
<mws xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" type="USERINFO" version="2.0">
  <userinfo>
    <attributes>
      <attribute name="UserId" value="TestUser" />
    </attributes>
  </userinfo>
</mws>
```

```
    <attribute name="UserRoleNames" value="Administrator_Role" />
  </attributes>
  <objectpermissions>
    <objectpermission objecttype="DataProviderDefinition"
objectactionpermission="Create Read Update Delete" />
    <objectpermission objecttype="AliasTable"
objectactionpermission="Create Read Update Delete" />
    <objectpermission objecttype="TextComponent"
objectactionpermission="Create Read Update Delete" />
    <objectpermission objecttype="TextComponent"
objectactionpermission="Create Read Update Delete" />
    <objectpermission objecttype="Enclosure"
objectactionpermission="Create Read Update Delete" />
    <objectpermission objecttype="TextDocumentCollection"
objectactionpermission="Create Read Update Delete" />
    <objectpermission objecttype="DataObject"
objectactionpermission="Create Read Update Delete" />
    <objectpermission objecttype="TextDocument"
objectactionpermission="Create Read Update Delete" />
    <objectpermission objecttype="EMailPrinter"
objectactionpermission="Create Read Update Delete" />
    <objectpermission objecttype="Form" objectactionpermission="Create
Read Update Delete" />
    <objectpermission objecttype="Envelope"
objectactionpermission="Create Read Update Delete" />
    <objectpermission objecttype="DataObjectQuery"
objectactionpermission="Create Read Update Delete" />
    <objectpermission objecttype="DataObjectScript"
objectactionpermission="Create Read Update Delete" />
    <objectpermission objecttype="MwsProcess"
objectactionpermission="Create Read Update Delete" />
    <objectpermission objecttype="LookupTable"
objectactionpermission="Create Read Update Delete" />
    <objectpermission objecttype="Printer"
objectactionpermission="Create Read Update Delete" />
    <objectpermission objecttype="Folder" objectactionpermission="Create
Read Update Delete" />
    <objectpermission objecttype="OutsourcingPrinter"
objectactionpermission="Create Read Update Delete" />
    <objectpermission objecttype="PostageDefinition"
objectactionpermission="Create Read Update Delete" />
    <objectpermission objecttype="Profile"
objectactionpermission="Create Read Update Delete" />
    <objectpermission objecttype="Process"
objectactionpermission="Create Read Update Delete" />
    <objectpermission objecttype="Role" objectactionpermission="Create
```

```

Read Update Delete" />
    <objectpermission objecttype="RoleMapping"
objectactionpermission="Create Read Update Delete" />
    <objectpermission objecttype="Selection"
objectactionpermission="Create Read Update Delete" />
    <objectpermission objecttype="Script" objectactionpermission="Create
Read Update Delete" />
    <objectpermission objecttype="StandardProcess"
objectactionpermission="Create Read Update Delete" />
    <objectpermission objecttype="SimpleTextContainer"
objectactionpermission="Create Read Update Delete" />
    <objectpermission objecttype="TransferJob"
objectactionpermission="Create Read Update Delete" />
    <objectpermission objecttype="ValueHelp"
objectactionpermission="Create Read Update Delete" />
    <objectpermission objecttype="CustomUI"
objectactionpermission="Create Read Update Delete" />
    <objectpermission objecttype="XsdSelection"
objectactionpermission="Create Read Update Delete" />
</objectpermissions>
<permissions>
    <permission>Administrate</permission>
    <permission>CommonSettings</permission>
    <permission>MwsInsertBlock</permission>
    <permission>MwsClearCache</permission>
    <permission>MwsDataSelection</permission>
    <permission>MwsInsertDocument</permission>
    <permission>MwsCreateDocuments</permission>
    <permission>CanPrintInPreview</permission>
    <permission>MwsPrintAndArchive</permission>
    <permission>MwsSetBundlePrinter</permission>
    <permission>MwsSetDocumentPrinter</permission>
    <permission>MwsSetDocumentPrinterWizzard</permission>
    <permission>MwsDownloadDocument</permission>
    <permission>MwsLoadExternalBlockFile</permission>
    <permission>MwsLoadExternalDocumentFile</permission>
    <permission>MwsAdministrate</permission>
    <permission>MwsViewOpenEnvelope</permission>
    <permission>MwsViewProcess</permission>
    <permission>MwsViewStack</permission>
    <permission>MwsViewOpenJob</permission>
    <permission>MwsShowNavigator</permission>
    <permission>MwsChangeFolder</permission>
    <permission>MwsChangeSystem</permission>
    <permission>ExecuteModusStudio</permission>
    <permission>ShowNavigator</permission>

```

```
<permission>ShowTypeFilter</permission>
<permission>OdinJobRemove</permission>
<permission>OdinJobSetPrinter</permission>
<permission>OdinJobSetState</permission>
<permission>OdinStackEnvelopeRemove</permission>
<permission>OdinProcessSuspend</permission>
<permission>OdinProcessUnlock</permission>
<permission>OdinProcessDelete</permission>
<permission>OdinProcessSetBack</permission>
<permission>OdinStackSetBackToStreaming</permission>
<permission>OdinStackEdit</permission>
<permission>OdinStackUnlock</permission>
<permission>OdinStackFree</permission>
<permission>OdinStackDelete</permission>
<permission>OdinStackLock</permission>
<permission>OdinOpenJobCreateStack</permission>
<permission>OdinProcessCreateStack</permission>
<permission>OdinStackSetBack</permission>
<permission>OdinStackSetBackTo</permission>
<permission>ChangeFolder</permission>
<permission>MwsManualForwardProcess</permission>
<permission>MwsProcessList</permission>
<permission>MwsProcessListFromDefinedUser</permission>
<permission>ModifyFontFace</permission>
<permission>ModifyFontColour</permission>
<permission>ModifyFontStyleBold</permission>
<permission>ModifyFontStyleItalic</permission>
<permission>ModifyFontStyleUnderline</permission>
<permission>ModifyFontSize</permission>
<permission>Search</permission>
<permission>ChangeSystem</permission>
<permission>SynchronizeSystem</permission>
<permission>TransferSystem</permission>
<permission>Test</permission>
<permission>ModifyFontAlignment</permission>
<permission>MwsForwardedProcessList</permission>
</permissions>
</userinfo>
</mws>
```

## Funktionen

### ClearCache

Löscht den Cache des RepositoryRuntimeServices.

### Syntax

```
ClearCache(string sessionId, out string xmlResult )
```

Parameter	Beschreibung
sessionId	Gültiger Session Identifier (SessionId)
xmlResult	Leer wenn ok, andernfalls MwsError

#### Rückgabewert

Wert	Beschreibung
0	OK
<>0	Fehler Siehe <a href="#">Fehlercodes</a>

#### Doc\_GetFile\_Mime

Ermöglicht das Herunterladen eines zuvor auf dem Server erstellten Dokuments. Dieses wird dabei als base64 codierter String zurückgegeben.

#### Syntax

```
Doc_GetFile_Mime(string sessionId, string processId, string fileName, out string doc, out string xmlResult )
```

Parameter	Beschreibung
sessionId	Gültiger Session Identifier (SessionId)
processId	Id des Prozesses, dessen BusinessObjekt die Datei verwaltet.
fileName	Name der Datei, die zurückgegeben werden soll
doc	Base64 codierter String, der den Inhalt der Datei widerspiegelt.
xmlResult	Leer wenn OK

#### Rückgabewert

Wert	Beschreibung
0	OK
<>0	Fehler Siehe <a href="#">Fehlercodes</a>

## Doc\_SetFile\_Mime

Lädt eine Datei (base64 codierter String) auf den Server hoch.

### Syntax

```
Doc_SetFile_Mime(string sessionId, string processId, string fileName, string doc, out string xmlResult)
```

Parameter	Beschreibung
sessionId	Gültiger Session Identifier (SessionId)
processId	Id des Prozesses, dessen BusinessObjekt die Datei verwaltet.
fileName	Name der Datei, deren Inhalt geschrieben werden soll.
doc	Base64 codierter String, der den Inhalt der Datei widerspiegelt.
xmlResult	Leer wenn OK

### Rückgabewert

Wert	Beschreibung
0	OK
<>0	Fehler Siehe <a href="#">Fehlercodes</a>

## Login

Verwenden Sie diese Funktion, um sich am Server anzumelden. Die Anmeldung muss einmal beim Starten einer Sitzung durchgeführt werden und ist für die Dauer dieser Sitzung gültig.

### Syntax

```
Login(string userName, string password, string code, string options, out
string sessionID, out string xmlResult )
```

Parameter	Beschreibung
userName	Benutzername
password	Passwort im Klartext. Dieser Parameter wird nur verwendet, wenn der Parameter <code>code</code> eine leere Zeichenkette enthält.
code	Verschlüsseltes Passwort, das mit Encoder.exe erstellt wurde.
options	Das Login-XML, mit der Sie einen Alias-Benutzernamen übergeben können, der dann verwendet wird, wenn die Sitzung eine neue Kommunikation erstellt.  Weitere Informationen finden Sie unter <a href="#">Login XML</a> .  Um diese Funktionalität zu verwenden, müssen Sie auch die MWS-Konfiguration anpassen. Weitere Informationen finden Sie unter dem Element <code>serviceuser</code> in "Das Element <code>mws</code> in <code>Composer.MWS.exe.config</code> " im Handbuch <i>Content Composer Konfiguration</i> .
sessionID	Session Identifier, dieser gibt die Sitzungs-ID zurück, die für alle nachfolgenden Aufrufe verwendet werden muss.
xmlResult	Bei erfolgreichem Login das entsprechende <a href="#">UserInfo XML</a> , sonst das <a href="#">Error XML</a> .

### Rückgabewert

Wert	Beschreibung
0	OK
<>0	Fehler Siehe <a href="#">Fehlercodes</a>

### Logout

Diese Funktion wird zum Abmelden einer Sitzung verwendet.

**Hinweis:** Eine Sitzung muss immer mit einem Logout beendet werden, um sie auf dem Server zu löschen. Wird diese Regel nicht eingehalten, dann bleiben die Datensätze in der entsprechenden Tabelle stehen.

### Syntax

```
Logout(string sessionId, out string xmlResult)
```

Parameter	Beschreibung
sessionId	Gültige SessionId der Session, die abgemeldet werden soll.
xmlResult	Leer wenn ok, andernfalls MwsError

#### Rückgabewert

Wert	Beschreibung
0	OK
<>0	Fehler Siehe <a href="#">Fehlercodes</a>

#### Obj\_GetStructure

Gibt die Struktur des referenzierten Objekts zurück.

#### Syntax

```
int Obj_GetStructure(string sessionId, string processId, string rootRef, int maxLevel, out string xmlResult)
```

Parameter	Beschreibung
sessionId	Gültiger Session Identifier (SessionId)
processId	Id des Prozesses, dessen Objekt referenziert wird.
rootRef	Die Objekt Referenz innerhalb des Prozesses
maxLevel	-1: Alles 0: Nur das Objekt <> 0: Die Struktur mit allen Kindern auf der n-ten Ebene
xmlResult	Objekt XML, abhängig von der Komponente

#### Rückgabewert

Wert	Beschreibung
0	OK
<>0	Fehler Siehe <a href="#">Fehlercodes</a>

### Obj\_SetStructure

Es besteht die Möglichkeit Eigenschaften von bereits bestehenden Objekten zu setzen bzw. können über Kommandos Aktionen am Bündel/Dokument angewendet werden. Weitere Informationen finden Sie unter [Eingebettete Aktionen im Object XML](#).

#### Syntax

```
Obj_SetStructure(string sessionId, string processId, string content, string rootRef, int maxLevel, out string xmlResult)
```

Parameter	Beschreibung
sessionId	Gültiger Session Identifier (SessionId)
processId	Id des Prozesses, dessen Objekt referenziert wird.
content	BusinessObjekt-Informationen, die gesetzt werden sollen.
rootRef	Referenz des Objektes, dessen Struktur nach dem Aufruf zurückgegeben werden soll.
maxLevel	Bestimmt wie viele Ebenen zurückgegeben werden sollen
xmlResult	Wenn OK, Struktur des BusinessObjekts.

#### Rückgabewert

Wert	Beschreibung
0	OK
<>0	Fehler Siehe <a href="#">Fehlercodes</a>

## Obj\_Toggle

Schaltet den Status eines Objektes um und gibt die geänderte Struktur zurück.

Die Funktion liefert deshalb die geänderte Struktur zurück, nachdem auch die entsprechend abhängigen Toggle-Aktionen ausgeführt wurden.

**Hinweis:** Das Toggeln kann aufgrund von Gruppen und Aktionen dazu führen, dass auch andere Objekte automatisch getoggelt werden.

### Syntax

```
Obj_Toggle(string sessionId, string processId, string objRef, string rootRef, int maxLevel, out string xmlResult)
```

Parameter	Beschreibung
sessionId	Gültiger Session Identifier (SessionId)
processId	Id des Prozesses, dessen Objekt referenziert wird.
objRef	Referenz des umzuschaltenden Objekts
rootRef	Referenz des Objektes, dessen Struktur nach dem Aufruf zurückgegeben werden soll.
maxLevel	Bestimmt wie viele Ebenen zurückgegeben werden sollen
xmlResult	Bei Erfolg, Struktur des Objekts

### Rückgabewert

Wert	Beschreibung
0	OK
<>0	Fehler Siehe <a href="#">Fehlercodes</a>

## Ping

Gibt eine Standard Nachricht für Testzwecke zurück.

### Syntax

```
Ping()
```

### Rückgabewert

Standardnachricht **MWS Process Service**.

## Process\_Close

Schließt einen Prozess mit dem übergebenen Status.

### Syntax

```
Process_Close(string sessionId, string processId, int status, string options, out
string xmlResult)
```

Parameter	Beschreibung
sessionId	Gültiger Session Identifier (SessionId)
processId	Id des Vorgangs, der geschlossen werden soll.
status	<p><b>Status, der gesetzt werden soll</b></p> <p><b>0:</b> Vorgang speichern Vorgänge mit dem Status 0 können wieder in Bearbeitung genommen werden.</p> <p><b>1:</b> Der Vorgang ist in Bearbeitung. Der Wert 1 wird programmintern verwendet und sollte nicht übergeben werden, da dann ein Fehler erzeugt wird.</p> <p><b>2:</b> Der Vorgang wurde erfolgreich abgeschlossen. Der Dienst MWS_Assistant löscht automatisch Vorgänge, die mit diesem Status gespeichert wurden.</p> <p><b>3:</b> Der Vorgang war fehlerhaft, z.B. wegen Abbruch durch den Benutzer. Fehlerhafte Vorgänge werden vom Dienst MWS_Assistant nicht automatisch gelöscht. Der Content Composer Client löscht die durch den Benutzer abgebrochenen Vorgänge explizit.</p> <p><b>Andere Werte</b> Der Vorgang ist beendet.</p>
options	<p>Options XML für das Schließen des Prozesses mit Angabe von Titel und Beschreibung.</p> <p>Beispiel Xml für das Setzen des Titels und der Beschreibung:</p> <pre>&lt;?xml version="1.0" encoding="utf-16"?&gt; &lt;mws&gt;   &lt;process&gt;     &lt;title&gt;My Title&lt;/title&gt;     &lt;description&gt;My description&lt;/description&gt;   &lt;/process&gt; &lt;/mws&gt;</pre>
xmlResult	Leer wenn OK

### Rückgabewert

Wert	Beschreibung
0	OK
<>0	Fehler Siehe <a href="#">Fehlercodes</a>

## Process\_Create

Diese Funktion erzeugt einen neuen MWS Process.

Dabei werden folgende Aktionen auf dem Server ausgeführt:

- Laden des konfigurierten Prozesses, der den MWS-Ablauf definiert, aus dem Repository (falls nicht bereits geschehen).
- Anlegen eines Verzeichnisses für die Daten dieses Dokument-Erstellungsprozesses.
- Anlegen eines Datensatzes für diesen Prozess in der DB-Tabelle MWS\_Processes.
- Wurde ein Options-Xml übergeben (Parameter options), so wird dieses ausgewertet.
- Das Skript OnSetOptionsScript des MWS-Prozesses wird- falls vorhanden – ausgeführt.
- Falls vorhanden wird das Skript OnNewProcess des MWS-Prozess ausgeführt.
- Wurde im Options-Xml ein Start-Kommando übergeben, so wird dieses ausgeführt.

Tritt während der Ausführung dieser Methode ein Fehler auf, werden die dabei angelegten Prozess-Daten (Dateien und DB-Datensatz) wieder gelöscht.

### Syntax

```
Process_Create(string sessionId,ref string processId, string mSystem,string mType,string mTitle, string objectIndex,string objectName, string objectType,string objectData, string options,out string xmlResult)
```

Parameter	Beschreibung
sessionId	Gültiger Session Identifier (SessionId)
processId	Wird die ProcessId nicht übergeben, wird eine neue erzeugt.
mSystem	SystemOld, in der das zu ladende Objekt liegt.
mType	Freier Text, mit dem der Prozess-Typ angegeben werden kann.
mTitle	Freier Text, mit dem ein Titel für den Prozess angegeben werden kann.
objectIndex	Index des zu ladenden Objekts (Wenn dieser Parameter angegeben wird, wird der Name

Parameter	Beschreibung
	ignoriert)
objectName	Name des zu ladenden Objekts. Dieser Parameter wird ignoriert, wenn der objectIndex angegeben wurde.
objectType	Typ des zu ladenden Objekts (z.Z. nur P = Bündel)
objectData	Optionale XML Daten für ein Bündel. Die Daten werden der ersten passiven Selektion im Skript OnNewStart zugewiesen.
options	Options Xml für den MWS-Prozess.
xmlResult	Wenn der Prozess erfolgreich angelegt wurde, wird entweder die Bündelstruktur zurückgegeben, oder die Prozessinformationen. Die Bündelstruktur wird zurückgegeben, wenn ein Start Command in den Optionen mit übergeben wurde.

### Rückgabewert

Wert	Beschreibung
0	OK
<>0	Fehler Siehe <a href="#">Fehlercodes</a>

### Process\_Delete

Löschen eines offenen Vorgangs und dessen Server-Ressourcen.

#### Syntax

```
Process_Delete(string sessionId, string processId, out string xmlResult)
```

Parameter	Beschreibung
sessionId	Gültiger Session Identifier (SessionId)
processId	Id des zu löschenden Prozesses
xmlResult	Leer wenn ok, andernfalls MwsError

### Rückgabewert

Wert	Beschreibung
0	OK
<>0	Fehler Siehe <a href="#">Fehlercodes</a>

## Process\_Forward

Diese Funktion leitet einen Vorgang weiter

### Syntax

```
Process_Forward(string sessionId, string processId, string destUser, string destUsergroup, string options, out string xmlResult)
```

Parameter	Beschreibung
sessionId	Gültiger Session Identifier (sessionId)
processId	Id des Vorgangs, der weitergeleitet werden soll.
destUser	Name des Zielbenutzers, an den der Vorgang weitergeleitet werden soll.
destUsergroup	Name der Zielgruppe, an die der Vorgang weitergeleitet werden soll.
options	Schließ-Optionen XML. (Titel und Beschreibung)
xmlResult	Leer wenn OK

### Rückgabewert

Wert	Beschreibung
0	OK
<>0	Fehler Siehe <a href="#">Fehlercodes</a>

## Process\_GetInfo

Gibt Informationen über den Prozess zurück.

### Syntax

```
Process_GetInfo(string sessionId,string processId, string infoType,string options, out string xmlResult)
```

Parameter	Beschreibung
sessionId	Gültiger Session Identifier (SessionId)
processId	Id des Vorgangs, dessen Informationen zurück gegeben werden sollen.
infoType	Kommaseparierter String, der die Informationen angibt, die zurückgegeben werden sollen. Diese sind zum einen vordefiniert: PROCESSINFO POOLVARS und zum anderen MWS-prozessabhängig und werden über das OnGetInfo Skript ausgewertet und bestückt.
options	Wird derzeit nicht unterstützt.
xmlResult	ProcessInfo XML, wenn OK

### Rückgabewert

Wert	Beschreibung
0	OK
<>0	Fehler Siehe <a href="#">Fehlercodes</a>

### Process\_GetLastError

Ermittelt den letzten Fehler eines Vorgangs.

### Syntax

```
Process_GetLastError(string sessionId,string processId, out string xmlResult)
```

Parameter	Beschreibung
sessionId	Gültiger Session Identifier (SessionId)
processId	Id des Prozesses, dessen Fehler zurück gegeben werden soll.
xmlResult	Bei Erfolg wird das letzte Error Xml zurückgegeben.

**Rückgabewert**

Wert	Beschreibung
0	OK
<>0	Fehler Siehe <a href="#">Fehlercodes</a>

**Process\_GetList**

Gibt die Liste der aktuell vorhanden Prozesse zurück.

**Syntax**

```
Process_GetList(string sessionId,int startAt,int max, string options,out string xmlResult)
```

Parameter	Beschreibung
sessionId	Gültiger Session Identifier (SessionId)
startAt	Start-Index
max	Maximale Anzahl der Prozesse, die zurückgegeben werden sollen.
options	Mit diesem Parameter kann angegeben werden, welche Vorgänge in der zurückgegebenen Vorgangliste enthalten sein sollen.  <b>Mögliche Einstellungen</b> <ol style="list-style-type: none"> <li>1. Rückgabe der vom Benutzer selbst erzeugten Vorgänge (default bei Übergabe eines Leerstrings).</li> <li>2. Rückgabe der Vorgänge die an den Benutzer direkt oder an eine Benutzergruppe, der der Benutzer angehört, weitergeleitet wurden.</li> <li>3. Rückgabe der eigenen sowie der weitergeleiteten Vorgänge. Wird für diesen Parameter ein Leerstring übergeben, so werden nur die vom Benutzer selbst erzeugten Vorgänge (siehe Punkt 1) zurückgegeben.</li> </ol>
xmlResult	Bei Erfolg ein MwsProcessList-XML, sonst ein MwsError.

**Rückgabewert**

Wert	Beschreibung
0	OK
<>0	Fehler Siehe <a href="#">Fehlercodes</a>

## Process\_Open

Nimmt die Bearbeitung eines Vorgangs wieder auf und setzt den Benutzernamen des bearbeitenden Benutzers auf den aktuellen Benutzer.

Über die Optionen kann eingestellt werden, ob es möglich ist gesperrte Prozesse zu öffnen (Prozessübernahme).

### Syntax

```
Process_Open(string sessionId, string processId, string options, out string xmlResult)
```

Parameter	Beschreibung
sessionId	Gültiger Session Identifier (SessionId)
processId	VorgangId, die geöffnet werden soll.
options	Mit diesem Parameter kann eingestellt werden, ob vor dem Öffnen eines Prozesses eine Statusprüfung durchgeführt werden soll. Siehe XML Beispiel <a href="#">Optionen beim Öffnen eines Vorgangs</a> . <b>True:</b> Es können nur gespeicherte Prozesse (Status = 0) geöffnet werden. <b>False (default):</b> Es können auch gesperrte Prozesse (Status = 1) geöffnet und dadurch übernommen werden. Abgeschlossene Prozesse (Status = 2) können generell nicht geöffnet werden.
xmlResult	Wenn OK, dann die Struktur des Business-Objekts.

### Rückgabewert

Wert	Beschreibung
0	OK

Wert	Beschreibung
<>0	Fehler Siehe <a href="#">Fehlercodes</a>

### Process\_SetData

Setzt XML Daten für das BusinessObjekt.

#### Syntax

```
Process_SetData(string sessionId, string processId, string dataType, string data, out string xmlResult)
```

Parameter	Beschreibung
sessionId	Gültiger Session Identifier (SessionId)
processId	Id des Vorgangs, für den Daten gesetzt werden sollen
dataType	Name, dem diese Daten zugeordnet werden sollen. Aktuell handelt es sich um den Namen der DataReference.
data	XML Daten, die gesetzt werden sollen.
xmlResult	Leer wenn OK

#### Rückgabewert

Wert	Beschreibung
0	OK
<>0	Fehler Siehe <a href="#">Fehlercodes</a>

### Process\_Start

Startet eine Vorgangsaktion.

#### Syntax

```
Process_Start(string sessionId, string processId, string command, string options, out string xmlResult)
```

Parameter	Beschreibung
sessionId	Gültiger Session Identifier (SessionId)
processId	Vorgangs- ID des Vorganges dessen Aktion gestartet werden soll.
command	In diesem Parameter ist das auszuführende Kommando zu übergeben. Die Namen der Kommandos ergeben sich aus den ScriptInvoke Activities im MWS-Process.
options	In diesem Parameter können verschiedene Optionen des Vorgangs gesetzt werden. Hier können die Aktivitäten des Vorgangs gesetzt, Variablen übergeben werden etc.
xmlResult	<p>Bei Erfolg die Struktur des Bündels</p> <p><b>Beispiel</b></p> <pre>&lt;?xml version="1.0" encoding="utf-16"?&gt; &lt;mws type="OBJECT"&gt;   &lt;process id="b9187257-812a-4e88-afa6-facb621dd2eb"&gt;     &lt;object title="ApplicationPack" type="P" name="ApplicationPack" index="02062008-822-tdc-itl1" reference=""&gt;       &lt;object title="Partner_ApplicationPack" type="D" name="Partner_ApplicationPack" index="02062008-823-tdr-itl1" reference="10DE6C09-79CF-4547-97AA-6302CF976185" state="1" enabled="1" print_asktime="0" form="DIN_A4_80g/qm" printer="" printer_type="" printtime="2" print_changeable="0" copycount="0" copy="0" password="composer" filename="Partner ApplicationPack_10DE6C09-79CF-4547-97AA-6302CF976185_O.docx" /&gt;       &lt;object title="pagenum" type="D" name="pagenum" index="23062009-591-tdr-dm" reference="A7DE4DF9-00A4-41FC- 8A84-05BB1D9D893D" state="2" enabled="1" print_asktime="0" form="DIN_A4_80g/qm" printer="" printer_type="" printtime="2" print_changeable="1" copycount="0" copy="0" password="" filename="pagenum_A7DE4DF9-00A4-41FC-8A84-05BB1D9D893D_O.docx" /&gt;       &lt;object title="RepCopy" type="D" name="Partner_ ApplicationPack" index="02062008-824-tcr-itl1" reference="3C331B86-8552-4AD6-A242-29106BFBCA21" state="2" enabled="1" print_asktime="0" form="DIN_A4_80g/qm" printer="" printer_type="" printtime="2" print_changeable="0" copycount="0" copy="1" password="composer" copytext="Copy for Representative" /&gt;       &lt;object title="AgentCopy" type="D" name="Partner_ ApplicationPack" index="02062008-825-tcr-itl1"</pre>

Parameter	Beschreibung
	<pre>reference="2572D52B-D2EC-4864-81A3-A4017CD16F53" state="2" enabled="1" print_asktime="0" form="DIN_A4_80g/qm" printer="FinePrint" printer_type="OP" printtime="1" print_ changeable="1" copycount="0" copy="1" password="composer" copytext="Copy for Agent" /&gt;   &lt;/object&gt; &lt;/process&gt; &lt;/mws&gt;</pre>

### Rückgabewert

Wert	Beschreibung
0	OK
<>0	Fehler Siehe <a href="#">Fehlercodes</a>

### Rep\_CreateItem

Erzeugt ein Objekt des angegebenen Typs.

Der Benutzer benötigt die Berechtigung zum Anlegen des angegebenen Objekttyps.

### Syntax

```
Rep_CreateItem(string sessionId, string systemName, string objectName, string
objectType, out string xmlResult)
```

Parameter	Beschreibung
sessionId	Gültiger Session Identifier (sessionId)
systemName	SystemOld
objectName	Name des anzulegenden Objekts.
objectType	Typ des anzulegenden Objekts.

Parameter	Beschreibung
xmlResult	XML-serialisiertes MwsItemInfo XML Objekt <ul style="list-style-type: none"> <li>• Leer wenn für das angegebene Objekt nicht vorhanden ist.</li> <li>• Enthält im Fehlerfall die Fehlermeldung.</li> </ul>

### Rückgabewert

Wert	Beschreibung
0	OK
<>0	Fehler Siehe <a href="#">Fehlercodes</a>

### Rep\_CreateVersion

Erzeugt eine neue Version des angegebenen Objekts.

Der Benutzer benötigt die Berechtigung zum Anlegen des angegebenen Objekttyps.

Zum Erzeugen einer Releaseversion benötigt der Benutzer die entsprechende Berechtigung.

### Syntax

```
Rep_CreateVersion(string sessionId, string systemName, string objectName, string objectType, string versionType, string versionLabel, out string xmlResult)
```

Parameter	Beschreibung
sessionId	Gültiger Session Identifier (SessionId)
systemName	SystemOld
objectName	Name des anzulegenden Objekts. Wenn angegeben, muss auch der objectType angegeben werden.
objectType	Typ des anzulegenden Objekts. Wenn angegeben, muss auch der objectName angegeben werden.
versionType	Typ der neuen Version. Mögliche Werte: <ul style="list-style-type: none"> <li>• "ReleaseVersion"</li> <li>• "WorkVersion"</li> </ul>

Parameter	Beschreibung
versionLabel	Label, das der neuen Version zugewiesen wird.
xmlResult	XML-serialisiertes MwsItemInfo XML Objekt <ul style="list-style-type: none"> <li>• Leer wenn für das angegebene Objekt nicht vorhanden ist.</li> <li>• Enthält im Fehlerfall die Fehlermeldung.</li> </ul>

**Rückgabewert**

Wert	Beschreibung
0	OK
<>0	Fehler Siehe <a href="#">Fehlercodes</a>

**Rep\_GetBinFile\_Mime**

Diese Methode gibt die Blob Daten eines Objekts zurück.

**Syntax**

```
Rep_GetBinFile_Mime(string sessionId, string systemName, string options, string
objectId, string objectName, string objectType, out string doc, out string
xmlResult)
```

Parameter	Beschreibung
sessionId	Gültiger Session Identifier (SessionId)
systemName	SystemOld des zugrundeliegenden Systems
options	Dieser Parameter wird zur Zeit ignoriert und sollte immer als Leerstring übergeben werden.
objectId	ObjectId der zu ermittelnden Objekt Blob Daten. Wenn nicht angegeben dann muss der ObjectName und ObjectType angegeben werden.
objectName	Name des Objektes. Wenn ObjectName angegeben wurde, dann muss der Typ angegeben werden.

Parameter	Beschreibung
objectType	Der Typ der zu ermittelnden Objekt Daten ( z.B. D,B)
doc	Der Inhalt des Blob-Feldes als Base64 codierter String
xmlResult	Leer wenn Ok, ansonsten Mws Error Xml.

### Rückgabewert

Wert	Beschreibung
0	OK
<>0	Fehler Siehe <a href="#">Fehlercodes</a>

### Rep\_GetDataProviderDefinition

Liest eine DataProviderDefinition aus dem Repository.

#### Syntax

```
Rep_GetDataProviderDefinition(string sessionId, string systemName, string
dataProviderDefinitionId, string dataProviderDefinitionName, out string
xmlResult)
```

Parameter	Beschreibung
sessionId	Gültiger Session Identifier (SessionId)
systemName	SystemOld
dataProviderDefinitionId	ObjektId der DataproviderDefinition. Wenn nicht angegeben, muss der Name der DataProviderDefinition angegeben werden.
dataProviderDefinitionName	Name der DataProviderDefinition
xmlResult	Xml-serialisiertes MwsDataProviderDefinition Objekt

### Rückgabewert

Wert	Beschreibung
0	OK
<>0	Fehler Siehe <a href="#">Fehlercodes</a>

## Rep\_GetFolderContent

Gibt den Inhalt eines Ordners zurück.

### Syntax

```
Rep_GetFolderContent(string sessionId, string systemName, string options, string filter, string folderName, out string xmlResult)
```

Parameter	Beschreibung
sessionId	Gültiger Session Identifier (SessionId)
systemName	SystemOld des zugrundeliegenden Systems
options	Derzeit nicht verwendet
filter	Filter Einstellungen für die Abfrage (D,P,B,A)
folderName	Folder DBID oder Folder Name
xmlResult	MwsFolderContent Xml serialisiert, oder MwsError im Fehlerfall.

### Rückgabewert

Wert	Beschreibung
0	OK
<>0	Fehler Siehe <a href="#">Fehlercodes</a>

## Rep\_GetForms

Ermittelt die Liste der Formulare aus dem Repository.

Wird eine DruckerOld oder ein DruckerName angegeben, werden nur die Formulare zurückgegeben, die der angegebene Drucker verwendet.

Bei Angabe der DruckerOld wird der Parameter `DruckerName` nicht ausgewertet.

### Syntax

```
Rep_GetForms(string sessionId, string systemName, string
usedByPrinterId, string usedByPrinterName, string printerType, out string
xmlResult)
```

Parameter	Beschreibung
sessionId	Gültiger Session Identifier (SessionId)
systemName	SystemOld des zugrundeliegenden Systems
usedByPrinterID	optional: PrinterOID
usedByPrinterName	optional: DruckerName, wird nur verwendet, wenn DruckerOld nicht belegt ist.
printerType	Optional: erforderlich wenn usedByPrinterName belegt ist. Mögliche Werte sind "OEP" für Email-Drucker, "OP" für Online-Drucker.
xmlResult	Formularliste als MwsFolderContent Xml Serialisiert, oder MwsError im Fehlerfall.

### Rückgabewert

Wert	Beschreibung
0	OK
<>0	Fehler Siehe <a href="#">Fehlercodes</a>

## Rep\_GetItemDescription

Gibt den Beschreibungstext eines Content Composer Objekts zurück.

### Syntax

```
Rep_GetItemDescription(string sessionId,string systemName, string
objectId,string objectName, string objectType,out string xmlResult)
```

Parameter	Beschreibung
sessionId	Gültiger Session Identifier (SessionId)
systemName	SystemOld des zugrundeliegenden Systems
objectId	ObjectId des Objektes, für das die ItemInfo gelesen werden soll. Wird dieser Parameter nicht angegeben, müssen der ObjectName und der ObjectType angegeben werden.
objectName	Name des Objektes, für das die Beschreibung gelesen werden soll. Wenn der ObjectName angegeben wurde, dann muss auch der ObjectTyp angegeben werden.
objectType	Typ des Objektes, für das die Beschreibung gelesen werden soll.
xmlResult	XML-serialisiertes MwsItemDescription-Objekt oder MwsError im Fehlerfall.

### Rückgabewert

Wert	Beschreibung
0	OK
<>0	Fehler Siehe <a href="#">Fehlercodes</a>

### Beispiel

```
<mws xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" type="ITEMDESCRIPTION"
version="2.0">
  <itemdescription>
    <description>This is the description of this text block</description>
    <objectname>MyTextBlock</objectname>
    <objectid>04062014-1744-tc-s</objectid>
    <databaseid>22306</databaseid>
    <objecttype>TextComponent</objecttype>
  </itemdescription>
</mws>
```

## Rep\_GetLookupObjects

Schlägt in einer Alias Tabelle die Werte zu einem Alias nach und interpretiert die Werte als Objektverweise (RepositoryReferences).

### Syntax

```
Rep_GetLookupObjects(string sessionId, string systemName, string options, string aliasTableId, string aliasTableName, string alias, out string xmlResult)
```

Parameter	Beschreibung
sessionId	Gültiger Session Identifier (SessionId)
systemName	SystemOld
options	Wird aktuell nicht ausgewertet.
aliasTableId	ObjektId der Aliastabelle Wird keine ObjectId angegeben, dann muss ObjektName und Typ angegeben werden.
aliasTableName	Name der Aliastabelle, in der der Alias nachgeschlagen werden soll.
alias	Alias, dessen Werte gelesen werden sollen.
xmlResult	Rückgabe der Objektverweise in Form eines MwsLookupResult Xml (beinhaltet MwsFolderContent Xml)

### Rückgabewert

Wert	Beschreibung
0	OK
<>0	Fehler Siehe <a href="#">Fehlercodes</a>

## Rep\_GetLookupValue

Gibt den Wert für den angegebenen Schlüssel aus der angegebenen Nachschlagetabelle zurück.

### Syntax

```
Rep_GetLookupValue(string sessionId, string systemName, string options, string lookupTableId, string lookupTableName, string key, out string xmlResult)
```

Parameter	Beschreibung
sessionId	Gültiger Session Identifier (SessionId)
systemName	SystemOld
options	Wird aktuell nicht ausgewertet.
lookupTableId	ObjektId der Nachschlagetabelle Wenn nicht angegeben, müssen ObjektName und Typ angegeben werden.
lookupTableName	Name der Nachschlagetabelle
key	Key
xmlResult	Rückgabe der Objektverweise in Form eines MwsLookupResult Xml (beinhaltet MwsLookupValue Xml)

#### Rückgabewert

Wert	Beschreibung
0	OK
<>0	Fehler Siehe <a href="#">Fehlercodes</a>

#### Rep\_GetNavigList

Ermittelt die Liste der Objekte eines Ordners in einem System.

Diese Funktion erfordert eine Pfadangabe und wertet den am Benutzer eingestellten Startordner aus.

Dieser wird intern automatisch am Anfang der übergebenen Pfadangabe angefügt.

#### Syntax

```
Rep_GetNavigList(string sessionId, string systemName, string options, string filter, string path, out string xmlResult)
```

Parameter	Beschreibung
sessionId	Gültiger Session Identifier (SessionId)

Parameter	Beschreibung
systemName	SystemOld
options	Dieser Parameter wird zur Zeit ignoriert und sollte immer als Leerstring übergeben werden.
filter	Kommaliste mit Typbezeichnungen der Objekte, die in der Rückgabe enthalten sein sollen. Beispiel: "A,P" (Ordner und Bündel)
path	DBId des Ordners, oder eine vollqualifizierte Pfadangabe (/Name/Name1/Name2)
xmlResult	Bei fehlerfreier Ausführung ein MwsNavigList Xml, ansonsten MwsError Xml.

### Rückgabewert

Wert	Beschreibung
0	OK
<>0	Fehler Siehe <a href="#">Fehlercodes</a>

### Rep\_GetNavigTree

Ermittelt den Verzeichnisbaum eines Systems.

#### Syntax

```
Rep_GetNavigTree(string sessionId, string systemName, string options, out string xmlResult)
```

Parameter	Beschreibung
sessionId	Gültiger Session Identifier (SessionId)
systemName	SystemOld des zugrundeliegenden Systems
options	Options XML, weitere Informationen finden Sie unter <a href="#">MwsNavigTree Options XML</a> .
xmlResult	<a href="#">MwsNavigTree-XML</a> wenn Ok, ansonsten <a href="#">Error XML</a> .

### Rückgabewert

Wert	Beschreibung
0	OK
<>0	Fehler Siehe <a href="#">Fehlercodes</a>

## Rep\_GetPrinters

Ermittelt die Liste der Online- und E-Mail-Drucker aus dem Repository.

Wird eine FormularOld oder ein FormularName angegeben, werden nur die Drucker zurückgegeben, die das angegebene Formular verwenden.

Bei Angabe der FormularOld wird der Parameter FormularName nicht ausgewertet.

### Syntax

```
Rep_GetPrinters(string sessionId, string systemName, string containingFormId, string containingFormName, out string xmlResult)
```

Parameter	Beschreibung
sessionId	Gültiger Session Identifier (SessionId)
systemName	SystemOld des zugrundeliegenden Systems
containingFormId	optional: ObjektId des Formulars
containingFormName	optional: optional: Name des Formulars (wird nur verwendet wenn keine FormularOld angegeben)
xmlResult	Druckerliste als MwsFolderContent Xml serialisiert, oder MwsError im Fehlerfall.

### Rückgabewert

Wert	Beschreibung
0	OK
<>0	Fehler Siehe <a href="#">Fehlercodes</a>

## Rep\_GetSystem

Ermittelt die Systeminformationen aus dem Repository.

### Syntax

```
Rep_GetSystem(string sessionId,string systemName, out string xmlResult)
```

Parameter	Beschreibung
sessionId	Gültiger Session Identifier (SessionId)
systemName	SystemOld des Systems, dessen Informationen zurückgegeben werden sollen.
xmlResult	SystemDef Xml wenn OK, ansonsten Mws Error Xml

### Rückgabewert

Wert	Beschreibung
0	OK
<>0	Fehler Siehe <a href="#">Fehlercodes</a>

## Rep\_GetSystemList

Gibt die Systemliste zurück.

### Syntax

```
Rep_GetSystemList(string sessionId,string options, out string xmlResult)
```

Parameter	Beschreibung
sessionId	Gültiger Session Identifier (SessionId)
options	Derzeit nicht verwendet
xmlResult	MwsSystemList xml

### Rückgabewert

Wert	Beschreibung
0	OK
<>0	Fehler Siehe <a href="#">Fehlercodes</a>

### Rep\_GetValueHelpDefinition

Liest eine Werthilfe aus dem Repository.

#### Syntax

```
Rep_GetValueHelpDefinition(string sessionId, string systemName, string valueHelpId, string valueHelpName, out string xmlResult)
```

Parameter	Beschreibung
sessionId	Gültiger Session Identifier (SessionId)
systemName	SystemOId des zugrundeliegenden Systems
valueHelpId	ObjectId der Werthilfe. Wenn nicht angegeben, muss der Name der Werthilfe angegeben werden.
valueHelpName	Name der Werthilfe
xmlResult	XML-serialisiertes MwsValueHelpDefinition Objekt

#### Rückgabewert

Wert	Beschreibung
0	OK
<>0	Fehler Siehe <a href="#">Fehlercodes</a>

### Rep\_LockItem

Sperrt das angegebene Objekt exklusiv.

#### Syntax

```
Rep_LockItem(string sessionId,string systemName,string objectId,string
objectName,string objectType,out string xmlResult)
```

Parameter	Beschreibung
sessionId	Gültiger Session Identifier (SessionId)
systemName	SystemOld
objectId	ObjectId, Wenn nicht angegeben, müssen objectName und objectType angegeben werden.
objectName	Name des zu sperrenden Objekts. Wenn angegeben, muss auch der objectType angegeben werden.
objectType	Typ des zu sperrenden Objekts. Wenn angegeben, muss auch der objectName angegeben werden.
xmlResult	XML-serialisiertes MwsItemInfo XML Objekt <ul style="list-style-type: none"> <li>• Leer wenn für das angegebene Objekt nicht vorhanden ist.</li> <li>• Enthält im Fehlerfall die Fehlermeldung.</li> </ul>

#### Rückgabewert

Wert	Beschreibung
0	OK
<>0	Fehler Siehe <a href="#">Fehlercodes</a>

#### Rep\_ReadItemBlob

Liest die Blob-Daten eines Objekts.

Liest die Blob-Daten eines Objekts.

#### Syntax

```
Rep_ReadItemBlob(string sessionId,string systemName,string objectId,string
objectName,string objectType,string blobType, out string blob, out string
xmlResult)
```

Parameter	Beschreibung
sessionId	Gültiger Session Identifier (sessionId)
systemName	SystemOld
objectId	ObjectId, Wenn nicht angegeben, müssen objectName und objectType angegeben werden.
objectName	Name des zu lesenden Objekts. Wenn angegeben, muss auch der objectType angegeben werden.
objectType	Typ des zu lesenden Objekts. Wenn angegeben, muss auch der objectName angegeben werden.
blobType	Typ des zu lesenden Blobs. Mögliche Werte: <ul style="list-style-type: none"> <li>• "WordDocument"</li> <li>• "XmlSchema"</li> <li>• "WordTemplate"</li> </ul>
blob	Base64 codierter String, der das Blob-Feld enthält.
xmlResult	XML-serialisiertes MwsItemInfo XML Objekt <ul style="list-style-type: none"> <li>• Leer wenn für das angegebene Objekt nicht vorhanden ist.</li> <li>• Enthält im Fehlerfall die Fehlermeldung.</li> </ul>

### Rückgabewert

Wert	Beschreibung
0	OK
<>0	Fehler Siehe <a href="#">Fehlercodes</a>

### Rep\_UnLockItem

Entsperrt das angegebene Objekt.

#### Syntax

```
Rep_UnLockItem(string sessionId, string systemName, string objectId, string objectName, string objectType, out string xmlResult)
```

Parameter	Beschreibung
sessionId	Gültiger Session Identifier (sessionId)
systemName	SystemOld
objectId	ObjectId, Wenn nicht angegeben, müssen objectName und objectType angegeben werden.
objectName	Name des zu entsperrenden Objekts. Wenn angegeben, muss auch der objectType angegeben werden.
objectType	Typ des zu entsperrenden Objekts. Wenn angegeben, muss auch der objectName angegeben werden.
xmlResult	XML-serialisiertes MwsItemInfo XML Objekt <ul style="list-style-type: none"> <li>• Leer wenn für das angegebene Objekt nicht vorhanden ist.</li> <li>• Enthält im Fehlerfall die Fehlermeldung.</li> </ul>

### Rückgabewert

Wert	Beschreibung
0	OK
<>0	Fehler Siehe <a href="#">Fehlercodes</a>

### Rep\_UpdateItemBlob

Aktualisiert einen bestimmten Blob eines Objekts.

Der Benutzer benötigt die Berechtigung zum Anlegen des angegebenen Objekttyps.

Wenn die Objekt-Versioning aktiv ist, ist die Aktualisierung nur für möglich, wenn das Objekt eine Work-Version ist.

### Syntax

```
Rep_UpdateItemBlob(string sessionId, string systemName, string objectId, string
objectName, string objectType, string blobType, string blob, out string
xmlResult)
```

Parameter	Beschreibung
sessionId	Gültiger Session Identifier (SessionId)
systemName	SystemOld
objectId	ObjectId, Wenn nicht angegeben, müssen objectName und objectType angegeben werden.
objectName	Name des zu aktualisierenden Objekts. Wenn angegeben, muss auch der objectType angegeben werden.
objectType	Typ des zu aktualisierenden Objekts. Wenn angegeben, muss auch der objectName angegeben werden.
blobType	Typ des zu aktualisierenden Blobs. Mögliche Werte: <ul style="list-style-type: none"> <li>• "WordDocument"</li> <li>• "XmlSchema"</li> <li>• "WordTemplate"</li> </ul>
blob	Base64 codierter String, der das Blob-Feld enthält.
xmlResult	XML-serialisiertes MwsItemInfo XML Objekt <ul style="list-style-type: none"> <li>• Leer wenn für das angegebene Objekt nicht vorhanden ist.</li> <li>• Enthält im Fehlerfall die Fehlermeldung.</li> </ul>

### Rückgabewert

Wert	Beschreibung
0	OK
<>0	Fehler Siehe <a href="#">Fehlercodes</a>

### Srv\_GetInfo

Gibt Information über den MWS Server zurück.

#### Syntax

```
Srv_GetInfo(string options, out string xmlResult)
```

Parameter	Beschreibung
options	Derzeit nicht verwendet.
xmlResult	MwsServerInfo XML wenn OK, andernfalls Mws Error XML

**Rückgabewert**

Wert	Beschreibung
0	OK
<>0	Fehler Siehe: <a href="#">Rückgabewerte von MWS Funktionen</a>

**Usr\_GetRoles**

Gibt alle Rollen aus dem konfigurierten RoleMapper zurück.

**Syntax**

```
Usr_GetRoles(string sessionId, out string lastError, out string[] roles)
```

Parameter	Beschreibung
sessionId	Gültiger Session Identifier (SessionId)
lastError	Fehlermeldung bzw. Leerstring bei erfolgreichem Aufruf.
roles	StringArray aller vorhandenen Rollen

**Rückgabewert**

Wert	Beschreibung
0	OK
-100	Fehler

**Usr\_GetUsers**

Gibt alle Benutzer zurück, die einer beliebigen Rolle aus dem RoleMapper angehören.

**Syntax**

```
Usr_GetUsers(string sessionId, out string lastError, out string[] users)
```

Parameter	Beschreibung
sessionId	Gültiger Session Identifier (SessionId)
lastError	Fehlermeldung bzw. Leerstring bei erfolgreichem Aufruf.
users	StringArray aller Benutzernamen, die einer beliebigen Rolle aus dem RoleMapper angehören.

#### Rückgabewert

Wert	Beschreibung
0	OK
-100	Fehler

#### Usr\_GetUsersOfRole

Gibt alle Benutzer zurück, die der angegebenen Rolle angehören.

#### Syntax

```
Usr_GetUsersOfRole(string sessionId, string roleName, out string lastError, out string[] users)
```

Parameter	Beschreibung
sessionId	Gültiger Session Identifier (SessionId)
roleName	Rolle, für die die Benutzer zurückgegeben werden sollen.
lastError	Fehlermeldung bzw. Leerstring bei erfolgreichem Aufruf.
users	StringArray aller Benutzernamen der angegebenen Rolle.

#### Rückgabewert

Wert	Beschreibung
0	OK
-100	Fehler

## WS UserRepository

Dieses Modul wird seit Version Foundation EP1 nicht mehr verwendet.

Die Funktionen dieses Moduls befinden sich nun im Modul [WS Process](#).

## Rückgabewerte von MWS Funktionen

Alle Funktionen des MWS Server geben im Regelfall eine Nummer zurück.

Diese Nummer sagt aus, ob die Funktion erfolgreich ausgeführt wurde oder nicht.

Grundsätzlich wird bei einem Code  $\neq 0$  in der XML Rückgabe ein Error Xml zurückgegeben. Dieses XML beschreibt detailliert den aufgetretenen Fehler.

### Ergebnis bei erfolgreicher Ausführung

**0:** Aufruf wurde erfolgreich verarbeitet

### Fehlercodes

Fehlercode	Beschreibung
-1	Fehler kann nicht eindeutig zugeordnet werden, Details in der XML Fehlermeldung
1	Fehler beim Funktionsaufruf, Details in der XML Fehlermeldung
302	Fehler beim Setzen von übergebenen Optionen
303	Fehler beim Setzen der Bündelstruktur
304	Fehler beim Schließen des Bündels
305	Fehler beim Initialisieren der Datenbeschaffung
309	Die maximale Anzahl der Client-Lizenzen wurde erreicht. <b>Dieser Rückgabecode wird erzeugt wenn:</b> <ul style="list-style-type: none"> <li>In der Server-Lizenz die maximale Anzahl der Clients die Anzahl der Sessions in der Sessiontabelle übersteigt.</li> <li>Keine Lizenz für den Server verfügbar ist. In diesem Fall ist nur eine Session erlaubt.</li> </ul>
311	Prozess ist abgeschlossen (Status = 2)
312	Prozess ist gesperrt (Status = 1)

<b>Fehlercode</b>	<b>Beschreibung</b>
314	Fehlende Berechtigung zum Anlegen eines Objekts vom angegebenen Typ.
315	Fehlende Berechtigung zum Update eines Objekts vom angegebenen Typ.
316	Fehlende Berechtigung zum Lesen eines Objekts vom angegebenen Typ
317	Es sind nicht alle notwendigen Parameter angegeben.
318	Das angegebene Item wurde nicht gefunden.
319	Der angegebene blobType ist ungültig oder nicht erlaubt.
320	Die Aktion ist für diesen versionType nicht erlaubt oder nicht möglich.
321	Der angegebene versionType ist ungültig.
322	Objektversionierung ist nicht aktiviert oder nicht lizenziert.
323	Fehlende Berechtigung zum Anlegen einer Release-Version.
324	Das Item ist durch einen anderen Benutzer gesperrt.
325	Item ist ReadOnly.
326	Item ist logisch gelöscht.
327	Ein Item mit diesem Namen und Typ existiert bereits.
328	Das Item ist gesperrt.
329	Das Item ist geschützt.
330	Der Objektname ist ungültig.
331	Systemzugriff verweigert.
332	Kein Wert gefunden für das angegebene AppSetting.
333	WordTemplate nicht gefunden.
334	Zurücksetzen von DocumentCompilationDate fehlgeschlagen.

Fehlercode	Beschreibung
335	Alias nicht in AliasTable gefunden.
336	LookupKey nicht in LookupTable gefunden.
337	Angefordertes Licensefeature nicht verfügbar.

## Warnungen

Warnungen liegen im Bereich von 6000 bis 6999.

**6000:** Aktion kann nicht ausgeführt werden, da weitere Konfigurationen benötigt werden.

## Informationen zur Integration

Die Web Services eröffnen nahezu unbegrenzte Integrationsmöglichkeiten.

Sowohl für die technische als auch die fachliche Integration der Korrespondenzerstellung stehen mit dieser Architektur sehr flexible Möglichkeiten zur Verfügung.

### Technische Integration

Die standardisierte Schnittstelle der Web Services ist der SOAP Interface Layer.

Dieser stellt somit die tiefste Integrationsmöglichkeit der Web Services dar. Jede Programmiersprache, die SOAP verarbeiten kann, lässt sich zur Programmierung von nativen Clients einsetzen.

Auf Basis dieser Schnittstelle lassen sich server- sowie clientseitige Integrationen implementieren.

Eine Integration in Webanwendungen kann natürlich auch über eine direkte Einbindung von z.B. Active Server Pages oder den Aufruf der Content Composer Webanwendung über Internetlinks erfolgen.

### Fachliche Integration

Der große Vorteil für die fachliche Integration eines Korrespondenzvorgangs über die Web Services liegt in der hohen Flexibilität.

Die Korrespondenzerstellung mit der Content Composer besteht aus einer Reihe von Aktivitäten, die miteinander kooperieren. Diese können entweder automatisiert ablaufen oder vom Benutzer manuell gestartet bzw. bearbeitet werden.

Die Reihenfolge der Ausführung dieser Aktivitäten ist bedingt durch Abhängigkeiten der Ergebnisse von Vorgängeraktivitäten.

Die Web Services erlauben einen granularen Zugriff auf diese einzelnen Aktivitäten.

Dadurch kann ohne weiteres auf unnötige Schritte verzichtet werden und es können eine oder mehrere Aktivitäten wiederholt werden. Das Konzept erlaubt also eine flexible Anpassung der Arbeitsabläufe an die fachlichen Anforderungen.

Beispielszenario Die Parametrisierung der Datenbeschaffung ist recht komplex und über mehrere Web-Seiten implementiert. Eine Dokumentauswahl, die Eingabe von manuellen Variablen und eine Nachbearbeitung sind nicht erwünscht. Es sollen nur Kontrolldokumente angezeigt und, wenn korrekt, archiviert werden.

Eine solches Szenario lässt sich über die Web Services ohne weiteres implementieren.

## Informationen zum Repository

Das Repository stellt die Grundlage für die Web Services dar.

Von hier werden alle Definitionen bezogen. Diese Definitionen können in unterschiedliche Bereiche aufgeteilt werden:

- Korrespondenzrelevante Daten wie Bündel, Textbausteine, Dokumentvorlagen etc.
- Daten aus dem Bereich Outputmanagement wie Drucker, Formulare etc.
- Systembezogene Daten

Über eine spezielle Schnittstelle lassen sich Informationen aus diesen Bereichen beziehen. Eine Drittanwendung hat damit z.B. die Möglichkeit eine Systemauswahl oder eine Auswahl von Bündeln incl. der Ordner-Hierarchie zu ermöglichen.

## Korrespondenzvorgänge

Ein Korrespondenzvorgang besteht aus mehreren Aktivitäten, die miteinander kooperieren. Diese können entweder automatisiert ablaufen oder vom Benutzer oder einer Drittanwendung manuell gestartet bzw. bearbeitet werden.

Aktivitäten können auch parallel ablaufen.

### **Verwaltung eines Korrespondenzvorgangs**

Ein Korrespondenzvorgang wird in einer Datenbanktabelle und im Dateisystem verwaltet und durch einen eindeutigen Schlüssel identifiziert. Der Schlüssel kann auch von Fremdsystemen vorgegeben werden.

Normalerweise beschränkt sich die Lebensdauer eines Korrespondenzvorgangs auf die Erstellung der Dokumente und deren Ausdruck. Unter dem Ausdruck der Dokumente wird hier die Übergabe an ein Outputmanagementsystem wie Odin verstanden.

Es besteht auch die Möglichkeit, einen Korrespondenzvorgang zu speichern, um ihn zu einem späteren Zeitpunkt fortzusetzen. Dies wird durch einen internen Status des Vorgangs ermöglicht.

### **Der Status eines Korrespondenzvorgangs**

Ein neuer Vorgang wird immer mit dem Status 1 (in Bearbeitung) angelegt.

Beim Schließen des Vorgangs kann entweder der Status 0 gesetzt werden (gespeichert, aber für weitere Bearbeitung offen) oder auch der Status 2 (Vorgang ist abgeschlossen und kann prinzipiell gelöscht werden).

### **Aktionen in einem Korrespondenzvorgang**

Folgende Aktionen gehören zu einem Korrespondenzvorgang:

- Anlegen des Korrespondenzvorgangs
- Festlegen der Parameter für die Datenbeschaffung (optional)
- Datenbeschaffung (optional)
- Anreichern der Bündelstruktur (dynamisches Bündel, optional)
- Konfigurieren der Dokumente (optional)
- Festlegen der manuellen Variablen (optional)
- Generieren der Dokumente
- Nachbearbeiten der Dokumente (optional)
- Drucken und Archivieren (optional)
- Schließen des Korrespondenzvorgangs