



*Der SAP-Conversion-Agent
von Itemfield*

Conversion Agent COM API-Referenz

Version 4

Conversion Agent COM API-Referenz

Copyright © 2004 - 2006 Itemfield Inc. Alle Rechte vorbehalten.

Itemfield hat möglicherweise Patente, angemeldete Patente, Marken, Urheberrechte oder sonstige Rechte an geistigem Eigentum inne, die Inhalte dieses Dokumentes abdecken. Sofern nichts anderes ausdrücklich in einem schriftlichen Lizenzvertrag mit Itemfield vereinbart wurde, erhalten Sie durch die Bereitstellung dieses Dokumentes keinerlei Anspruch auf diese Patente, Marken, Urheberrechte oder auf sonstiges geistiges Eigentum.

Die in diesem Dokument enthaltenen Informationen können jederzeit ohne vorherige Ankündigung geändert werden. Der Anwender trägt die Verantwortung für die Einhaltung aller anwendbaren Urheberrechte. Ohne ausdrückliche schriftliche Genehmigung durch Itemfield Inc. darf kein Teil dieses Dokumentes in irgendeiner Form oder auf irgendeine elektronische oder mechanische Weise zu irgendeinem Zweck vervielfältigt oder weitergegeben werden.

SAP AG
<http://www.sap.com>

Informationen zur Veröffentlichung:

Version: 4

Datum: September 2006

Inhalt

Überblick	1
Plattform-Unterstützung.....	1
Programmiersprachen-Unterstützung	1
Einführung	2
Wie Sie das API anwenden	2
Beispielanwendung	2
Online-Beispiel	5
Objekt-Modell-Referenz	5
CMDS-Objekt	6
CMRequest-Objekt	6
Input-Methoden	7
Output-Methoden	9
Methode zum Generieren einer Anfrage	11
CMEngine-Objekt	12
CMStatus-Objekt	13
Veraltete COM API-Versionen	15
CM_COM-Implementierung	15
CMParsingEngine-Objekt.....	16
CM_COM2-Implementierung.....	18

Überblick

Eine Anwendung kann einen Conversion-Agent-Dienst ausführen, indem sie das Conversion Agent COM API aufruft. Das API ist mehrprozessgestützt und wiedereintrittsfähig, und unterstützt somit High-Throughput-Anwendungen.

Das API aktiviert Conversion Agent Engine. Sie können damit:

- Die Namen eingesetzter Conversion-Agent-Dienste abrufen
- Den auszuführenden Dienst bestimmen
- Den Eingabeort oder -typ bestimmen (Pfad, URL, String, Datenstrom oder Buffer)
- Den Ausgabeort oder -typ bestimmen (Pfad, String oder Datenstrom)
- Den Dienst ausführen
- Den Ausführungsstatus abrufen (Return-Code, Fehlerbeschreibung usw.)

Plattform-Unterstützung

Das COM API läuft unter Microsoft Windows.

Wenn Sie ein API benötigen, das sowohl unter Windows als auch auf anderen Plattformen läuft, sollten Sie das Java API verwenden (siehe *Conversion Agent Java API-Referenz*).

Programmiersprachen-Unterstützung

Die Beispiele und Syntax in diesem Buch sind in Microsoft Visual Basic 6.0 geschrieben.

Sie können in jeder Sprache programmieren, die COM (ActiveX)-Unterstützung bietet, wie zum Beispiel Visual Basic.NET, Visual C++ usw. Sollten Sie Informationen benötigen, wie man das COM API in einer bestimmten Sprache anwendet, kontaktieren Sie bitte SAP-Support.

Einführung

Eine Einführung in das COM API finden Sie im Kapitel *Conversion Agent Engine ausführen* im Buch *Einführung in Conversion Agent*.

In diesem Kapitel wird das COM API zur Programmierung einer Visual Basic 6 Anwendung verwendet, die einen Dienst in Conversion Agent Engine ausführt. Es ist der komplette Quellcode angegeben.

Wie Sie das API anwenden

Der typische Ablauf für die Anwendung des Conversion Agent COM API sieht wie folgt aus:

1. Konfigurieren Sie in Conversion Agent Studio ein Projekt, das einen Parser, Serializer, Mapper oder Transformer ausführt.
2. Machen Sie das Projekt als Conversion-Agent-Dienst verfügbar.
Informationen dazu, wie Sie die Schritte 1 und 2 ausführen, finden Sie im *Conversion Agent Studio Anwenderhandbuch*.
3. Programmieren Sie eine Anwendung, die das COM API aufruft, um den Dienst auszuführen. Die Anwendung sollte folgende Schritte durchführen:
 - a. Sie können wahlweise das `CMDS`-Objekt verwenden, um die Namen der eingesetzten Conversion-Agent-Dienste abzurufen.
 - b. Verwenden Sie das `-Objekt`, um eine Ausführungsanfrage zu erstellen. Die Anfrage gibt den Namen des Dienstes sowie Eingabe-/Ausgabeort oder `-typ` an.
 - c. Verwenden Sie das `-Objekt`, um die Anfrage auszuführen.
 - d. Verwenden Sie das `-Objekt`, um den Ausführungsstatus, wie etwa Return-Code oder Fehlerinformationen, abzurufen.
4. Kompilieren Sie die Anwendung.
5. Installieren und führen Sie die Anwendung auf einem Windows-Host-Computer aus, auf dem Conversion Agent Engine installiert ist.

Beispielanwendung

Das folgende Beispiel ist eine Visual Basic 6.0-Funktion, die das COM API verwendet, um einen Conversion-Agent-Dienst auszuführen.

Die Funktion geht davon aus, dass Sie die Eingabe in einem Buffer bereitstellen. Sie generiert die Ausgabe als String. Die Parameter lauten:

Parameter	Typ	Ein/Aus	Beschreibung
strService	String	Ein	Name des Conversion-Agent-Dienstes.
varBufIn	Variant	Ein	Der Eingabe-Buffer (der zum Beispiel das Quelldokument eines Parsers enthält).
lngBufInLength	Long	Ein	Die Länge des Eingabebuffers in Byte.
strOutput	String	Aus	Die Ausgabe (zum Beispiel, die von einem Parser generierte XML). Wenn ein Fehler auftritt, enthält strOutput eine Fehlermeldung.
Rückgabewert	Integer	Aus	1, wenn der Dienst erfolgreich ausgeführt wurde.

Das Visual Basic-Projekt referenziert die CM_COM3 1.0 Type Library (CM_COM3.dll). Das ist die Conversion Agent COM API Bibliothek.

Quellcode

```
Function RunCMService( _
    strService As String, _
    varBufIn As Variant, _
    lngBufInLength As Long, _
    strOutput As String) _
    As Boolean

    Dim objCMEngine As CM_COM3Lib.CMEngine
    Dim objCMRequest As CM_COM3Lib.CMRequest
    Dim objCMStatus As CM_COM3Lib.CMStatus
    Dim strRequest As String
    Dim strStatus As String

    'Generate a request string
    Set objCMRequest = New CM_COM3Lib.CMRequest
    strRequest = objCMRequest.Generate( _
        strService, _
        objCMRequest.BufferInput(lngBufInLength), _
        objCMRequest.TextOutput, _
        "", "", "")

    'Initialize Conversion Agent Engine
    Set objCMEngine = New CM_COM3Lib.CMEngine
    strStatus = objCMEngine.InitEngine()
```

```
' If the initialization failed, return the status string to the caller
Set objCMStatus = New CM_COM3Lib.CMStatus
If objCMStatus.IsGood(strStatus) <> 1 Then
    strOutput = strStatus
    RunCMService = False
    Exit Function
End If

' Execute the request
strStatus = objCMEngine.Exec( _
    strRequest, varBufIn, "", strOutput)
' Return 1 for success or 0 for failure
If objCMStatus.IsGood(strStatus) = 1 Then
    RunCMService = True
Else
    ' If the service failed, return the status string to the caller
    strOutput = strStatus
    RunCMService = False
End If

' Termination
Set objCMRequest = Nothing
Set objCMEngine = Nothing
Set objCMStatus = Nothing

End Function
```

Gebufferte Eingabe

Zur Veranschaulichung haben wir die RunCMService-Funktion so gestaltet, dass sie gebufferte Eingaben akzeptiert. Dadurch unterstützt die Funktion binäre oder codierte Texteingaben.

Das folgende Beispiel zeigt, wie Sie die gebufferte Eingabe einer Visual Basic-Anwendung vorbereiten. In dem Beispiel wird ein Conversion-Agent-Dienst namens Parser1 ausgeführt. Wir gehen davon aus, dass Parser1 Eingaben akzeptiert, die auf der Systemcodeseite enthalten sind.

Der Code führt folgende Operationen aus:

1. Er weist die Eingabe dem String "abcdef120" zu.
2. Standardmäßig speichert Visual Basic Strings in Unicode-Codierung. Die StrConv-Funktion konvertiert die Eingabe in die Systemcodeseite.
3. Der Code ruft RunCMService auf.
4. Wird RunCMService erfolgreich ausgeführt, zeigt der Code die XML-Ausgabe an, die Parser1 generiert hat. Funktioniert RunCMService nicht, zeigt der Code eine Fehlermeldung an.

```
Dim strOutput As String
Dim lngBufInLength As Long
Dim inp As String
Dim BufIn As String

'Assign the input
inp = "abcdef120"
lngBufInLength = Len(inp)

'Convert the input from a Unicode representation to the system code page
BufIn = StrConv(inp, vbFromUnicode)

'Run the Conversion Agent service
If RunCMService("Parser1", BufIn, lngBufInLength, strOutput) Then
    'If the service succeeded, display its output
    MsgBox "Returned: " & strOutput
Else
    'If the service failed, display the error message
    MsgBox "Error: " & strOutput
End If
```

Online-Beispiel

Ein weiteres Beispiel für die Dateiein- und -ausgabe finden Sie in der COM API-Übung im Kapitel *Conversion Agent Engine ausführen* im Buch *Einführung in Conversion Agent*.

Der Visual Basic Quellcode und eine kompilierte *.exe-Datei dieses Beispiels stehen Online unter folgendem Pfad in Ihrem Conversion Agent Programmordner zur Verfügung.

Tutorials\Exercises\CMComApi Tutorial

Weitere Visual Basic-Beispiele finden Sie unter folgendem Pfad im Conversion-Agent-Ordner:

Samples\API\COM API

Objekt-Modell-Referenz

Das COM API ist in der Datei CM_COM3.dll definiert, die sich im Conversion-Agent-Programmordner befindet (standardmäßig c:\Programme\SAP\

ConversionAgent\bin). In den Visual Basic Projektreferenzen heißt die Datei CM_COM3 1.0 Type Library.

Die Bibliothek stellt vier COM-Objekte zur Verfügung:

CMDS

Liefert Informationen über Conversion-Agent-Dienste.

CMRequest

Generiert einen Anfragestring, der Anweisungen für Conversion Agent Engine enthält, wie etwa den auszuführenden Dienst und den Eingabe-/Ausgabeort.

CMEngine

Führt die Anfrage in Conversion Agent Engine aus.

CMStatus

Ruft Informationen über den Status der Conversion Agent Engine-Operationen ab, wie etwa den Return-Code und Fehlerbeschreibungen.

Die einzelnen Objekte und ihre Schnittstellen werden im Folgenden beschrieben.

CMDS-Objekt

Mit diesem Objekt können Sie auf den Conversion-Agent-Speicher zugreifen. Das Objekt zeigt die ICMS-Schnittstelle an.

GetServiceList-Methode

Function GetServiceList() As String

Diese Methode liefert die Namen aller eingesetzten Dienste.

Parameter	Typ	Ein/Aus	Beschreibung
Rückgabewert	String	Aus	Ein Unicode-String, der die Namen aller eingesetzten Dienste enthält. Die Namen sind durch Semikola getrennt.

CMRequest-Objekt

Dieses Objekt erstellt Anfragestrings, die Anweisungen für Conversion Agent Engine enthalten. Ein Anfragestring gibt Folgendes an:

- Den auszuführenden Conversion-Agent-Dienst
- Den Eingabeort oder das Eingabeformat
- Den Ausgabeort oder das Ausgabeformat

- Optional einen Anwendernamen und ein Passwort, das der Dienst verwendet, um auf einen HTTP-Server zuzugreifen, der eine Identifizierung verlangt.

Gehen Sie wie folgt vor, um das CMRequest-Objekt zu verwenden:

1. Rufen Sie eine der Input-Methoden auf, um den Eingabeort anzugeben.
Dieser Schritt ist für einen Dienst, der einen Parser ausführt, nicht zwingend vorgeschrieben. Wenn Sie diesen Schritt weglassen, müssen Sie die Eingabe in der sources_to_extract-Eigenschaft des Parsers definieren.
2. Rufen Sie eine der Output-Methoden auf, um den Ausgabeort anzugeben.
3. Geben Sie das Ergebnis der Input und Output-Methoden, gemeinsam mit anderen Parametern, wie dem Dienstenamen an die Generate-Methode weiter. Damit wird der Anfragestring generiert.
4. Geben Sie die Anfragespezifikation an das CMEngine-Objekt, das die Anfrage ausführt, weiter.

Das Objekt zeigt die ICMRequest-Schnittstelle an.

Input-Methoden

Das COM API unterstützt Eingaben aus einer Datei, einer URL, einem String, einem Buffer oder einem Datenstrom. Dazu müssen Sie die entsprechende Input-Methode aufrufen.

Wenn der Conversion-Agent-Dienst beispielsweise ein Quelldokument, das in einer Datei gespeichert ist, parsen soll, dann rufen Sie die FileInput-Methode auf. Wenn das Quelldokument von einem COM-Objekt zur Verfügung gestellt wird, das die IStream-Schnittstelle anzeigt, dann rufen Sie die IStreamInput-Methode auf.

Der Rückgabewert der Methoden ist ein Unicode-String, der eine Eingabespezifikation enthält.

FileInput-Methode

Function FileInput(fileName As String) As String

Diese Methode gibt eine Datei an, aus der ein Dienst seine Eingabe erhält.

Parameter	Typ	Ein/Aus	Beschreibung
fileName	String	Ein	Ein Unicode-String, der Pfad und Name der Eingabedatei enthält.
Rückgabewert	String	Aus	Die Eingabespezifikation

TextInput-Methode

Function TextInput(text As String) As String

Diese Methode gibt einen Unicode-String an (bis zu 2 kB), der als Eingabe für einen Dienst dient.

Das COM API übersetzt entsprechend der Systemcodeseite den Unicode in ANSI. Die Eingabecodierung des Conversion-Agent-Dienstes muss daher der Systemcodeseite entsprechen.

Wegen der besseren Leistung empfehlen wir, die `BufferInput`-Methode statt der `TextInput`-Methode zu verwenden.

Parameter	Typ	Ein/Aus	Beschreibung
text	String	Ein	Der Unicode Eingabestring.
Rückgabewert	String	Aus	Die Eingabespezifikation.

UrlInput-Methode

Function `UrlInput(url As String) As String`

Diese Methode gibt eine URL an, von der ein Dienst seine Eingabe erhält.

Parameter	Typ	Ein/Aus	Beschreibung
url	String	Ein	Ein Unicode-String, der die URL enthält.
Rückgabewert	String	Aus	Die Eingabespezifikation.

BufferInput-Methode

Function `BufferInput(size As Long) As String`

Diese Methode legt fest, dass ein Dienst seine Eingabe aus einem Buffer erhält.

Parameter	Typ	Ein/Aus	Beschreibung
size	Long	Ein	Die Buffergröße in Byte.
Rückgabewert	String	Aus	Die Eingabespezifikation.

IStreamInput-Methode

Function `IStreamInput() As String`

Diese Methode legt fest, dass ein Dienst seine Eingabe aus einem Datenstrom erhält (jedes beliebige Objekt, das die `IStream`-Schnittstelle anzeigt).

Parameter	Typ	Ein/Aus	Beschreibung
Rückgabewert	String	Aus	Die Eingabespezifikation.

Output-Methoden

Das COM API unterstützt Ausgaben in eine Datei, einen String oder einen Datenstrom. Dazu müssen Sie die entsprechende Output-Methode aufrufen.

Wenn der Conversion-Agent-Dienst beispielsweise seine Ausgabe in eine Datei schreiben soll, dann rufen Sie die `FileOutput`-Methode auf, um den Dateinamen festzulegen. Sie können mit dem Dateinamen einen absoluten Pfad festlegen oder Sie können eine der `OutputLocation... Methoden` aufrufen, um den Pfad anzugeben.

Der Rückgabewert der Methoden ist ein Unicode-String, der eine Ausgabespezifikation enthält.

FileOutput-Methode

Function `FileOutput(fileName As String) As String`

Diese Methode gibt eine Datei an, in die ein Dienst seine Ausgabe schreibt.

Parameter	Typ	Ein/Aus	Beschreibung
<code>fileName</code>	String	Ein	Ein Unicode-String, der den Namen der Ausgabedatei enthält. Der Name kann wahlweise einen absoluten oder relativen Pfad enthalten. Ein relativer Pfad wird in Relation zum Ordner, den Sie mit einer <code>OutputLocation... Methode</code> angeben, angelegt.
Rückgabewert	String	Aus	Die Ausgabespezifikation.

OutputLocation... Methoden

Wenn Sie die `FileOutput`-Methode aufgerufen haben, ohne einen absoluten Pfad anzugeben, rufen Sie eine der folgenden Methoden auf, um den Ordner für die Ausgabe festzulegen.

Function `OutputLocationCurrentFolder() As String`

Gibt an, dass Conversion Agent die Ausgabe im aktuellen Ordner speichern soll.

Function `OutputLocationResultFolder() As String`

Gibt an, dass Conversion Agent die Ausgabe im Results-Ordner des Projekts speichern soll.

Function OutputLocationSpecifyPath(Path as String) As String

Gibt an, dass Conversion Agent die Ausgabe unter dem angegebenen absoluten Pfad speichern soll.

Function OutputLocationGUIDFolder(Path as String) As String

Diese Methode entspricht OutputLocationSpecifyPath, abgesehen davon, dass die Ausgabe in einem vom System erstellten Unterordner des Pfades gespeichert wird. Der Name des Unterordners ist der Dienstname, dem ein GUID folgt. Die Ausgabe jeder Anfrage wird in einem eigenen Unterordner gespeichert.

Wenn Sie beispielsweise OutputLocationSpecifyPath("c:\temp") aufrufen, speichert Conversion Agent die Ausgabedatei im Ordner c:\temp.

Parameter	Typ	Ein/Aus	Beschreibung
Path	String	Ein	(Bei den Methoden OutputLocationSpecifyPath und OutputLocationGUIDFolder) Der Ausgabepfad.
Rückgabewert	String	Aus	Die Spezifikation des Ausgabeortes.

TextOutput-Methode

Function TextOutput() As String

Diese Methode legt fest, dass ein Dienst seine Ausgabe in einen String schreibt.

Parameter	Typ	Ein/Aus	Beschreibung
Rückgabewert	String	Aus	Die Ausgabespezifikation.

IStreamOutput-Methode

Function IStreamOutput() As String

Diese Methode legt fest, dass ein Dienst seine Ausgabe in einen Datenstrom schreibt (jedes beliebige Objekt, das die IStream-Schnittstelle anzeigt).

Parameter	Typ	Ein/Aus	Beschreibung
Rückgabewert	String	Aus	Die Ausgabespezifikation.

Methode zum Generieren einer Anfrage

Nachdem Sie die Input und Output-Methoden aufgerufen haben, rufen Sie die Generate-Methode auf.

Generate-Methode

```
Function Generate( _
    service As String, _
    input As String, _
    output As String, _
    outputLocation As String, _
    userName As String, _
    password As String) _
    As String
```

Diese Methode kombiniert die Ergebnisse der Input- und Output-Methoden mit anderen Parametern. Sie liefert einen Unicode-Anfragestring in dem von der Exec-Methode des CMEngine-Objekts verlangten Format.

Die Parameter `service` und `output` sind für alle Anfragen erforderlich.

Der Eingabeparameter ist für alle Anfragen erforderlich. (Wenn Sie ihn weglassen, wird der Dienst aufgrund der Eingabe ausgeführt, die in der Eigenschaft `example_source` des Dienstes definiert ist.)

Wenn die `example_source` des Dienstes einen Dokumentprozessor definiert, wendet der Dienst ihn auf die Eingabe an.

Die Parameter `userName` und `password` sind nützlich, wenn der Dienst auf einen Web-Server zugreift, der eine HTTP-Identifizierung verlangt. Wenn Sie vom Datentyp `non-empty` sind, heben Sie den in den Projekteigenschaften angegebenen Anwendernamen und das Passwort auf.

Parameter	Typ	Ein/Aus	Beschreibung
<code>service</code>	String	Ein	Ein Unicode-String, der den Namen des Dienstes enthält, den Conversion Agent Engine ausführt.
<code>input</code>	String	Ein	Die Eingabespezifikation (Rückgabewert einer Input-Methode).
<code>output</code>	String	Ein	Die Ausgabespezifikation (Rückgabewert der Methode <code>FileOutput</code> , <code>TextOutput</code> , oder <code>StreamOutput</code>).

Parameter	Typ	Ein/Aus	Beschreibung
outputLocati on	String	Ein	Die Pfadspezifikation für Ausgaben in eine Datei (Rückgabewert einer OutputLocati on... Methode).
userName	String	Ein	Ein Unicode-String, der einen Anwendernamen enthält, den ein Dienst für die HTTP-Identifizierung verwendet.
password	String	Ein	Ein Unicode-String, der ein Passwort enthält, das ein Dienst für die HTTP-Identifizierung verwendet.
Rückgabewert	String	Aus	Der Anfragestring.

CMEngine-Objekt

Das CMEngi ne-Objekt führt in Conversion Agent Engine eine Anfrage aus. Das Objekt zeigt die ICMEngi ne-Schnittstelle an.

InitEngine-Methode

Function Ini tEngi ne() As String

Diese Methode initialisiert die Conversion Agent Engine-Instanz.

Rufen Sie Ini tEngi ne auf, wenn Sie zum ersten Mal ein CMEngi ne-Objekt erstellen, bevor Sie Exec aufrufen. Sie sollten sicherstellen, dass Ini tEngi ne einen Erfolgsstatus liefert, andernfalls ist nicht vorherzusehen, welche Ausgabe Exec liefert.

Parameter	Typ	Ein/Aus	Beschreibung
Rückgabewert	String	Aus	Ein Unicode-String, der einen Statusbericht enthält.

Exec-Methode

Function Exec(_
 request As String, _
 input As Variant, _
 output As Variant, _
 outputStr As String) _
 As String

Diese Methode führt die Dienst-Anfrage aus.

Parameter	Typ	Ein/Aus	Beschreibung
request	String	Ein	Der Anfrage-String (der Rückgabewert der Methode CMRequest.Generate).
input	Variant	Ein	Bei Buffereingaben: das Buffer-Objekt (in C++ ein Zeiger auf den Buffer) Bei Datenstromeingaben: das IStream-Objekt (in C++ ein Zeiger auf das IStream-Objekt). Bei anderen Eingabetypen: ignoriert.
output	Variant	Aus	Bei Datenstromausgaben: das IStream-Objekt (in C++ ein Zeiger auf das IStream-Objekt). Bei anderen Ausgabetypen: ignoriert.
outputStr	String	Aus	Bei Textausgaben: der Ausgabestring. Bei anderen Ausgabetypen: ein Statusstring.
Rückgabewert	String	Aus	Ein Unicode-String, der einen Statusbericht enthält.

CMStatus-Objekt

Das CMStatus-Objekt ruft Statusinformationen über Conversion Agent Engine-Operationen ab.

Die Eingabe für jede Methode ist ein Statusbericht, der von CMEngine generiert wurde (der Rückgabewert einer CMEngine-Methode). Die Ausgabe ist die geforderte Statusinformation.

Das Objekt zeigt die ICMStatus-Schnittstelle an.

IsGood-Methode

Function IsGood(statusStr As String) As Long

Diese Methode liefert den Wert 1 (wahr), wenn die Conversion Agent Engine-Operation erfolgreich durchgeführt wurde, oder den Wert 0 (falsch), wenn sie nicht erfolgreich war.

Parameter	Typ	Ein/Aus	Beschreibung
statusStr	String	Ein	Der CMEngine Statusbericht.
Rückgabewert	Long (In IDL: boolean)	Aus	1, wenn erfolgreich.

GetDescription-Methode

Function GetDescription(statusStr As String) As String

Wenn die Conversion Agent Engine Operation nicht erfolgreich war, liefert diese Methode eine Fehlerbeschreibung.

Parameter	Typ	Ein/Aus	Beschreibung
statusStr	String	Ein	Der CMEngine Statusbericht.
Rückgabewert	String	Aus	Ein Unicode-String, der die Beschreibung enthält.

GetSeverity-Methode

Function GetSeverity(statusStr As String) As Long

Diese Methode liefert einen Wert, der angibt, wie schwerwiegend ein Fehler ist.

Parameter	Typ	Ein/Aus	Beschreibung
statusStr	String	Ein	Der CMEngine Statusbericht.
Rückgabewert	Long	Aus	Der Fehlergrad: 1, wenn bei der Ausführung nur Information-Ereignisse auftraten 2, wenn Warning-Ereignisse auftraten 3, wenn Failure-Ereignisse auftraten 4, wenn Fatal-error-Ereignisse auftraten 5, wenn Internal-error-Ereignisse auftraten

GetLogFile-Methode

Function GetLogFile(statusStr As String) As String

Konnte die Conversion Agent Engine-Operation nicht erfolgreich ausgeführt werden, liefert diese Methode den Namen der Protokolldatei.

Unter COM API generiert Conversion Agent Engine eine Protokolldatei, wenn während der Ausführung des Dienstes ein Fehler auftrat oder dieser nicht ausführbar war. Es generiert kein Protokoll, wenn der Dienst erfolgreich ausgeführt wurde. Dieses Verhalten ist unabhängig von den Protokolleinstellungen in den Projekteigenschaften.

Parameter	Typ	Ein/Aus	Beschreibung
statusStr	String	Ein	Der CMEngine Statusbericht.

Parameter	Typ	Ein/Aus	Beschreibung
Rückgabewert	String	Aus	Der Name der Protokolldatei.

GetLogDirectory-Methode

Function GetLogDirectory(statusStr As String) As String

Konnte die Conversion Agent Engine-Operation nicht erfolgreich ausgeführt werden, liefert diese Methode den Namen des Protokolldateiverzeichnisses.

Parameter	Typ	Ein/Aus	Beschreibung
statusStr	String	Ein	Der CMEngine Statusbericht.
Rückgabewert	String	Aus	Das Verzeichnis der Protokolldatei.

Veraltete COM API-Versionen

In den vorangehenden Abschnitten dieses Dokuments wird die aktuelle Version des Conversion Agent COM API, CM_COM3 1.0 Type Library, beschrieben, die in der Datei CM_COM3.dll implementiert ist. Die CM_COM3-Implementierung ist gegenüber den früheren COM API-Implementierungen verbessert und leistungsstärker.

Es gibt zwei frühere Versionen des COM API, die weiterhin unterstützt werden, sodass sie zu bestehenden Anwendungen abwärts kompatibel sind. In diesem Kapitel werden die früheren COM API-Versionen kurz beschrieben und Upgrade-Empfehlungen gegeben.

Beachten Sie bitte, dass Sie die COM API-Implementierungen nicht mischen können. Versuchen Sie beispielsweise nicht, eine Dienstanfrage vorzubereiten, indem Sie Methoden der CM_COM3-Implementierung aufrufen und die Anfrage dann ausführen, indem Sie die CM_COM-Implementierung aufrufen.

CM_COM-Implementierung

Conversion Agent 3.1 und frühere Versionen boten eine Implementierung des COM API, die CM_COM 1.0 Type Library genannt wurde. Die Implementierung wurde in der Datei CM_COM.dll definiert. Die Implementierung war nicht

mehrprozessgestützt und funktionierte deshalb nicht so gut wie die CM_COM3-Implementierung.

Wir empfehlen, dass Sie CM_COM-Anwendungen auf CM_COM3 upgraden. Ist das nicht machbar, können Sie die CM_COM-Anwendungen weiterhin ausführen.

CMParsingEngine-Objekt

Die CM_COM-Implementierung besteht aus einem einzigen Objekt, nämlich CMParsingEngine. Das Objekt führt Dienste in Conversion Agent Engine aus.

Das Objekt zeigt die ICMParsingEngine-Schnittstelle an.

Um das Objekt zu verwenden, machen Sie Folgendes:

1. Weisen Sie eine der Input-Eigenschaften zu, die den Ort der Eingabe angibt.
2. Rufen Sie eine der Methoden auf, die einen Conversion-Agent-Dienst ausführt, der die Eingabe verarbeitet.

InputFile-Eigenschaft

(String) Pfad und Name der Eingabedatei.

InputText-Eigenschaft

(String) Ein Textstring, der die Eingabe enthält.

InputURL-Eigenschaft

(String) Die URL einer Eingabedatei.

AuthUserName-Eigenschaft

(String) Ein Anwendername zur HTTP-Identifizierung.

AuthPassword-Eigenschaft

(String) Ein Passwort zur HTTP-Identifizierung.

ParseToFile-Methode

```
Function ParseToFile( _  
    service_name As String, _  
    output_file As String, _  
    error As Long) _  
    As String
```

Diese Methode führt einen Parser-Dienst aus und schreibt die XML-Ausgabe in eine Datei.

Parameter	Typ	Ein/Aus	Beschreibung
service_name	String	Ein	Name des Conversion-Agent-Dienstes.

Parameter	Typ	Ein/Aus	Beschreibung
output_file	String	Ein	Pfad der Ausgabedatei.
error	Long	Aus	Der Ausführungsstatus: 1, wenn bei der Ausführung nur Information-Ereignisse auftraten 2, wenn Warning-Ereignisse auftraten 3, wenn Failure-Ereignisse auftraten 4, wenn Fatal-error-Ereignisse auftraten
Rückgabewert	String	Aus	Ein Statusstring.

ParseToText-Methode

```
Function ParseToText( _
    service_name As String, _
    error As Long) _
    As String
```

Diese Methode führt einen Parser-Dienst aus und schreibt die XML-Ausgabe in einen String.

Parameter	Typ	Ein/Aus	Beschreibung
service_name	String	Ein	Der Name des Conversion-Agent-Dienstes.
error	Long	Aus	Der Ausführungsstatus: 1, wenn bei der Ausführung nur Information-Ereignisse auftraten 2, wenn Warning-Ereignisse auftraten 3, wenn Failure-Ereignisse auftraten 4, wenn Fatal-error-Ereignisse auftraten
Rückgabewert	String	Aus	Der Ausgabestring.

SerializeToFile-Methode

```
Function SerializeToFile( _
    service_name As String, _
    output_file As String, _
    error As Long) _
    As String
```

Diese Methode führt einen Serializer-Dienst aus und schreibt die Ausgabe in eine Datei. Die Parameter sind identisch mit jenen von ParseToFile.

SerializeToText-Methode

```
Function SerializeToText( _  
    service_name As String, _  
    error As Long) _  
    As String
```

Diese Methode führt einen Serializer-Dienst aus und schreibt die Ausgabe in einen String. Die Parameter sind identisch mit jenen von ParseToText.

CM_COM2-Implementierung

Conversion Agent 3.1 bot die CM_COM2 1.0 Type Library, die in der Datei CM_COM2.dll definiert war.

Die Implementierung war eine frühe Version von CM_COM3. Das Objekt-Modell ist dem von CM_COM3 ähnlich, aber einige Methoden wurden überarbeitet.

Wir empfehlen dringend, dass Sie alle bestehenden CM_COM2-Anwendungen auf CM_COM3 upgraden.