



Installation und Benutzung AD.NAV.MailTools

Version 1.0.1.0

ALTENBRAND Datentechnik GmbH

Am Gelicht 5

35279 Neustadt (Hessen)

Tel: 06692/202 290

Fax: 06692/204 741

eMail: support@altenbrand.de

Die Komponente AD.NAV.MailTools für Microsoft Dynamics NAV ermöglicht es Ihnen, aus Microsoft Dynamics NAV heraus eMails per SMTP zu versenden. Folgende Features werden hierbei unterstützt:

- Beliebig viele Empfänger
- Beliebig viele CC- und BCC-Empfänger
- Dateianhänge (Warten auf Erzeugung der Datei möglich)
- STARTTLS-Verschlüsselung
- Beliebige Encoding des Mail-Inhalts
- Plaintext- oder HTML-Inhalt
- Ersetzung von Platzhaltern durch dynamische Inhalte im Mail-Text

Voraussetzungen

Voraussetzung für den Einsatz der Komponente ist die Installation des Microsoft .NET Framework 4.0. Sollte das Microsoft .NET Framework auf dem Zielsystem noch nicht installiert sein, können Sie das benötigte Installationsprogramm unter folgender Adresse kostenlos herunterladen:

<http://www.microsoft.com/de-de/download/details.aspx?id=17718>

Installation

Hintergrund

Microsoft Dynamics NAV 2009 und früher

In Microsoft Dynamics NAV 2009 und früher kann die Komponente ausschließlich als COM-Objekt verwendet werden. Im Rahmen der Ausführung des Installationsassistenten findet bereits die notwendige Registrierung der DLL im System statt. Hierbei ist zu beachten, dass die DLL am Installationsort nicht gelöscht werden darf, da das System ansonsten die Komponente nicht verwenden kann.

Als Besonderheit in Microsoft Dynamics NAV ist es jedoch so, dass sowohl die DLL- als auch die TLB-Datei in die Ordner der jeweiligen Microsoft Dynamics Bestandteile kopiert werden muss, in denen sie verwendet werden soll.

Wollen Sie die Komponente beispielsweise in CodeUnits verwenden, welche von einem Application Server aufgerufen werden, so müssen die DLL- und TLB-Datei in den Ordner dieses Application Servers kopiert werden. Soll die entsprechende CodeUnit jedoch auch aus dem Classic Client oder RTC aufgerufen werden, so müssen Sie die Dateien auch in die Ordner des Classic Client/RTC kopieren.

Für jeden Client-Rechner müssen Sie den Installationsassistenten ausführen, damit die Registrierung der COM-Komponente im System erfolgt, bevor Sie die Dateien in den Classic Client/RTC-Ordner kopieren.

Ab Microsoft Dynamics NAV 2013

Ab Microsoft Dynamics NAV 2013 ist eine Verwendung der Komponente direkt über den Datentyp „DotNet“ möglich. Die Registrierung als COM-Komponente im System erfolgt zwar dennoch, ist jedoch für diese Versionen nicht von Belang.

Die DLL muss jedoch in verschiedene Ordner kopiert werden:

- 1) Auf dem Rechner, der den Dienst „Microsoft Dynamics NAV Server“ ausführt, in den „Add-Ins“ Ordner des Programmverzeichnisses des Microsoft Dynamics NAV Server
- 2) Auf jedem Client-Rechner in den Ordner „Add-Ins“ des RTC.

Ablauf

Die Installation erfolgt wie unter Windows üblich mit Hilfe eines Installationsassistenten. Standardmäßig wird als Installationsordner der Ordner des Microsoft Dynamics NAV 2009 Classic Client im System gesucht und ggf. verwendet. Sie können jedoch auch jeden beliebigen anderen Ordner wählen, auf den Sie Zugriff haben.

Bitte beachten Sie, dass es je nach Einsatzgebiet sein kann, dass Sie die DLL- und ggf. auch die TLB-Datei noch in weitere Ordner kopieren müssen, damit die Komponente funktioniert. Folgende Ordner kommen unter anderem hierfür in Frage:

- **Verwendung im Classic Client von Microsoft Dynamics NAV 2009:**
z.B. C:\Program Files (x86)\Microsoft Dynamics NAV\60\Classic
- **Verwendung im Role Tailored Client (RTC) von Microsoft Dynamics NAV 2009:**
z.B. C:\Program Files (x86)\Microsoft Dynamics NAV\60\RoleTailored Client
- **Verwendung im Dienst „Microsoft Dynamics NAV Server“ ab Microsoft Dynamics NAV 2013 auf Serverseite:**
Unterordner „Add-ins“ des „Microsoft Dynamics NAV Server“ oder ein Unterordner dieses Ordners
z.B. C:\Program Files\Microsoft Dynamics NAV\XY\Service\Add-ins\ALTENBRAND Datentechnik GmbH
- **Verwendung im Role Tailored Client (RTC) ab Microsoft Dynamics NAV 2013:**
Unterordner „Add-ins“ des RTC oder ein Unterordner dieses Ordners
z.B. C:\Program Files (x86)\Microsoft Dynamics NAV\XY\RoleTailored Client\Add-ins\ALTENBRAND Datentechnik GmbH

Einbindung in Microsoft Dynamics NAV

Je nach eingesetzter Version von Microsoft Dynamics NAV gibt es verschiedene Möglichkeiten, die Komponente einzubinden.

COM Interface (Datentyp AUTOMATION)

Die Komponente kann ab Microsoft Dynamics NAV 5.0 über eine COM-Schnittstelle (Component Object Model) verwendet werden. In Microsoft Dynamics NAV können Komponenten, welche eine COM-Schnittstelle bereitstellen, ganz einfach als „Automation Object“ eingebunden und angesprochen werden.

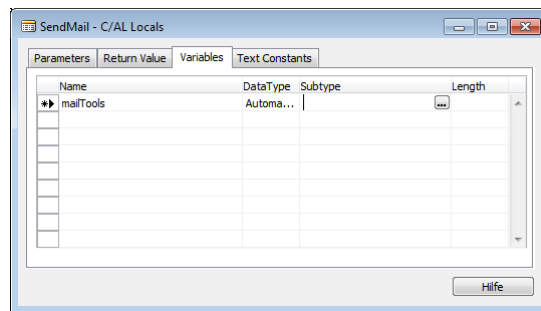
Im folgenden Beispiel wird in einer CodeUnit eine Funktion erstellt, welche eine eMail an einen Kunden sendet und eine Datei anhängt.

Erzeugen Sie zunächst eine CodeUnit und in dieser CodeUnit eine Function namens `SendMail`.

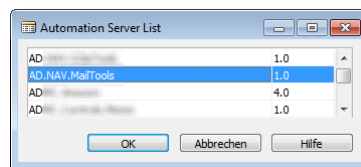


Legen Sie nun dieser Function zunächst mit „Ansicht > Locals“ zwei neue lokale Variablen für den Zugriff auf die Komponente an.

Die erste Variable ermöglicht den Zugriff auf die Komponente selbst. Geben Sie hierzu auf dem Reiter Variables den Namen `mailTools` ein und wählen Sie als DataType „Automation“.

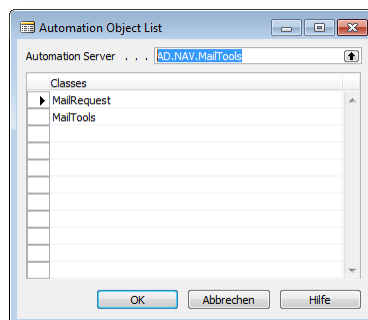


Wechseln Sie in das Feld SubType und drücken Sie die Taste F6. Nun öffnet sich die Maske „Automation Object List“, welche noch leer ist, da zunächst ein Automation Server gewählt werden muss. Drücken Sie hierzu wiederum F6 und öffnen Sie so die Maske „Automation Server List“. In dieser Maske werden alle im Betriebssystem registrierten COM-Komponenten angezeigt.



Suchen Sie die den Eintrag `AD.NAV.MailTools`, markieren Sie ihn mit der Maus und drücken Sie dann OK.

In der Maske „Automation Object List“ sehen Sie nun die bereitgestellten Automatisierungsobjekte. Wählen Sie hier die Class `MailTools`.



Die zweite Variable wird benötigt, um die Funktionsaufrufe durchzuführen und das Ergebnis des Mailversands zurückzuliefern. Verfahren Sie wie oben, nennen Sie die Variable jedoch `request` und wählen Sie in der „Automation Object List“ die Class `MailRequest`.

Abschließend geben Sie folgenden Quellcode für Funktion ein und speichern Sie dann die CodeUnit:

SendMail ()

```
// ACHTUNG: UNTERSCHIEDLICHE ERZEUGUNG IN CLASSIC CLIENT UND RTC!
// Erzeuge Automatisierungsobjekte (Classic Client)
CREATE(mailTools);
CREATE(request);

// Erzeuge Automatisierungsobjekte (RTC)
//CREATE(mailTools, TRUE, TRUE);
//CREATE(request, TRUE, TRUE);

// Betreff
request.Subject := 'Test für eine Mail';

// Absender
request.SenderMailAddress := 'test@test.de';
request.SenderName := 'Testbenutzer';

// Konfiguration des Inhalts
request.IsMailBodyHtml := TRUE;
request.BodyEncoding := 'utf-8';

// Vorlage verwenden und innerhalb der Vorlage einen Platzhalter ersetzen
request.MailTemplateFile := 'C:\MailTemplates\Sample.html';
request.AddVariable('%CUSTOMER%', 'Herr Mustermann');

// Eine PDF-Datei anhängen
request.AddAttachment('C:\Attachments\Sample.pdf');

// Empfänger hinzufügen
request.AddRecipient('max@mustermann.de');
request.AddCCRecipient('moritz@mustermann.de');

mailTools.SendMail('smtp.test.de', 25, 'benutzername', 'kennwort',
request);
IF request.RequestError THEN
    MESSAGE(request.RequestErrorMessage);

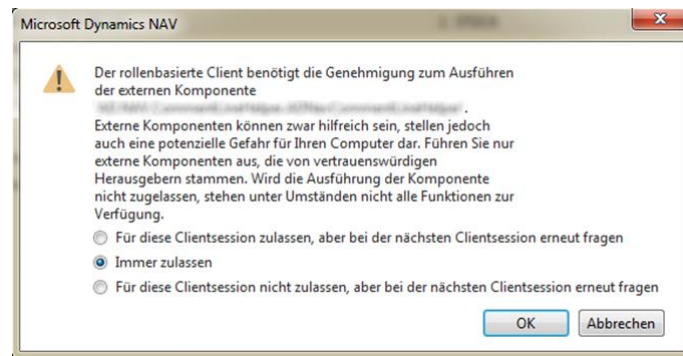
CLEAR(request);
CLEAR(mailTools);
```

Bitte beachten Sie, dass die Fehlerbehandlung im Programmierbeispiel darin besteht, einfach nur eine Meldung auszugeben und fortzufahren. Sollten Sie stattdessen die Ausführung bei Fehlern abbrechen wollen, müssen Sie auf jeden Fall dafür sorgen, dass die Automatisierungsobjekte freigegeben werden!

Wichtig: Im Unterschied zum Classic Client muss im Role Tailored Client festgelegt werden, ob das Automatisierungsobjekt auf dem Server oder auf dem Client erzeugt werden soll. Hierzu müssen im Aufruf von `CREATE` zwei zusätzliche Parameter mitgegeben werden (siehe Quellcode-Beispiel).

Ob das Automatisierungsobjekt auf dem Server erzeugt werden soll muss in Abhängigkeit davon entschieden werden, ob auch Dateien auf Anwender-PCs verarbeitet werden sollen. In diesem Fall ist es nicht möglich, ein Server-Automatisierungsobjekt zu verwenden.

Bei der ersten Verwendung der Funktionen müssen Sie eventuell bestätigen, dass die Funktionalität der Komponente innerhalb von NAV ausgeführt werden darf. Wählen Sie dazu die Option „Immer zulassen“ aus, und klicken Sie anschließend auf OK. Die Abfrage erscheint ggf. zweimal, weil zwei verschiedene Objekte verwendet werden (vgl. oben die Definition der lokalen Variablen).



.NET DLL (Datentyp DOTNET) ab NAV 2013

Die Komponente kann ab Microsoft Dynamics NAV 2013 über den Datentyp DotNet eingebunden und angesprochen werden.

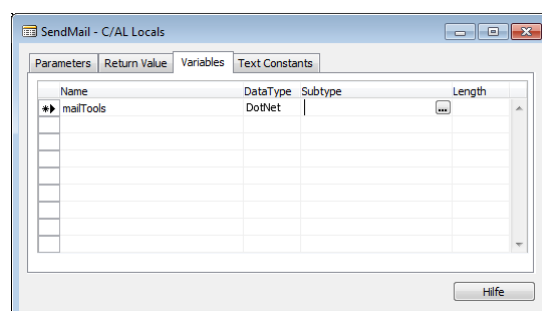
Im folgenden Beispiel wird in einer CodeUnit eine Funktion erstellt, welche eine eMail an einen Kunden sendet und eine Datei anhängt.

Erzeugen Sie zunächst eine CodeUnit und in dieser CodeUnit eine Function namens `SendMail`.

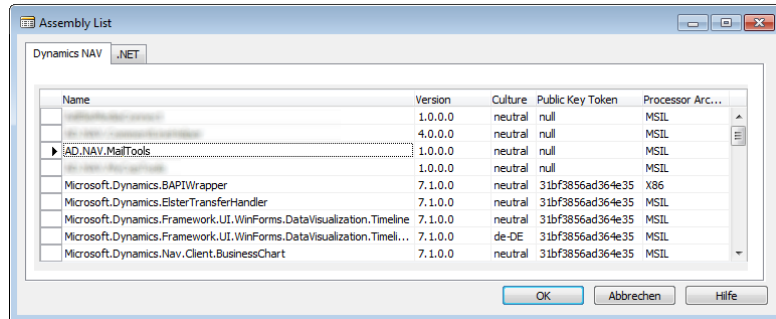


Legen Sie nun dieser Function zunächst mit „Ansicht > Locals“ zwei neue lokale Variablen für den Zugriff auf die Komponente an.

Die erste Variable ermöglicht den Zugriff auf die Komponente selbst. Geben Sie hierzu auf dem Reiter Variables den Namen `mailTools` ein und wählen Sie als DataType „DotNet“.

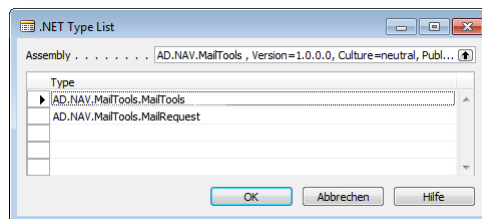


Wechseln Sie in das Feld SubType und drücken Sie die Taste F6. Nun öffnet sich die Maske „.NET Type List“, welche noch leer ist, da zunächst eine Assembly gewählt werden muss. Drücken Sie hierzu wiederum F6 und öffnen Sie so die Maske „Assembly List“. In dieser Maske werden die bekannten Assemblies angezeigt.

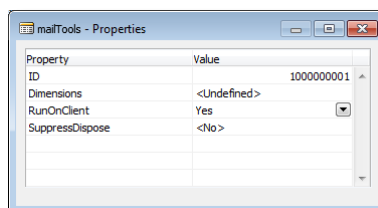


Suchen Sie im Reiter „Dynamics NAV“ den Eintrag AD.NAV.MailTools, markieren Sie ihn mit der Maus und drücken Sie dann OK.

In der Maske „.NET Type List“ sehen Sie nun die bereitgestellten .NET-Typen. Wählen Sie hier den Typ AD.NAV.MailTools.MailTools. Klicken Sie dann OK.



In den Properties für die lokale Variable mailTools setzen Sie anschließend noch die Property RunOnClient auf den Wert Yes.



Die zweite Variable wird benötigt, um die Funktionsaufrufe durchzuführen und das Ergebnis des Aufrufs zurückzuliefern. Verfahren Sie wie oben, nennen Sie die Variable jedoch request und wählen Sie in der „.NET Type List“ den .NET-Typ AD.NAV.MailTools.MailRequest. Vergessen Sie nicht, die Property RunOnClient auf den Wert Yes zu setzen.

Abschließend geben Sie folgenden Quellcode ein und speichern Sie dann die CodeUnit:

SendMail ()

```
// Erzeuge .NET Objekte
mailTools := mailTools.MailTools();
request := request.MailRequest();
```

```

// Betreff
request.Subject := 'Test für eine Mail';

// Absender
request.SenderMailAddress := 'test@test.de';
request.SenderName := 'Testbenutzer';

// Konfiguration des Inhalts
request.IsMailBodyHtml := TRUE;
request.BodyEncoding := 'utf-8';

// Vorlage verwenden und innerhalb der Vorlage einen Platzhalter ersetzen
request.MailTemplateFile := 'C:\MailTemplates\Sample.html';
request.AddVariable('%CUSTOMER%', 'Herr Mustermann');

// Eine PDF-Datei anhängen
request.AddAttachment('C:\Attachments\Sample.pdf');

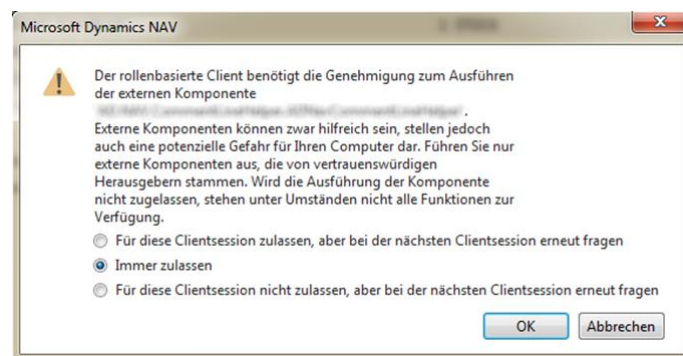
// Empfänger hinzufügen
request.AddRecipient('max@mustermann.de');
request.AddCCRecipient('moritz@mustermann.de');

mailTools.SendMail('smtp.test.de', 25, 'benutzername', 'kennwort',
request);
IF request.RequestError THEN
    MESSAGE(request.RequestErrorMessage);

CLEAR(request);
CLEAR(mailTools);

```

Bei der ersten Verwendung der Funktionen müssen Sie eventuell bestätigen, dass die Funktionalität der Komponente innerhalb von NAV ausgeführt werden darf. Wählen Sie dazu die Option „Immer zulassen“ aus, und klicken Sie anschließend auf OK. Die Abfrage erscheint ggf. zweimal, weil zwei verschiedene Objekte verwendet werden (vgl. oben die Definition der lokalen Variablen).



Konzepte, Funktionen und Eigenschaften

Ein Konzept für die Nutzung dieser Komponente ist, dass Funktionsaufrufe mit einem einzelnen Aufrufobjekt („RequestObject“) gesteuert werden. Dieses RequestObject ist sowohl in der Lage, alle benötigten Parameter für den Aufruf anzunehmen, als auch das Ergebnis des Aufrufs oder einen Fehlerzustand zurückzuliefern.

Funktionen der Komponente

Folgende Funktionen werden von der Komponente bereitgestellt:

Funktionsname	Typ des RequestObject	Beschreibung
SendMail	MailRequest	Sendet eine eMail.
GetTempPath	-	Liefert den im System konfigurierten temporären Ordner.

Funktion SendMail

Die Funktion SendMail erwartet folgende Parameter:

Typ	Name	VAR-Parameter	Bedeutung
BSTR	smtpServer	N	Der Name/die IP-Adresse des zu verwendenden SMTP-Servers.
LONG	smtpPort	N	Der Port, welcher für den Mailversand verwendet werden soll.
BSTR	smtpUser	N	Der Benutzername für den Zugriff auf den SMTP-Server.
BSTR	smtpPassword	N	Das Kennwort für den Zugriff auf den SMTP-Server.
_MAILREQUEST	request	J	Das RequestObject mit den Informationen über die zu versendende eMail.

Funktion GetTempPath

Die Funktion GetTempPath erwartet keine Parameter und liefert als Rückgabe einen Wert vom Typ BSTR, welcher den Pfadnamen enthält.

RequestObjects für die Funktionsaufrufe

MailRequest

Dieses RequestObject wird für die Funktion SendMail verwendet. Es ermöglicht die Übergabe der Informationen zur zu versendenden eMail.

Funktionen

AddAttachment

Fügt der eMail einen neuen Dateianhang hinzu. Dieser Funktionsaufruf wartet ggf. eine vorgegebene Zeitspanne, bis die Datei im System vorhanden ist (siehe Eigenschaften WaitForAttachmentToBeReady und WaitForAttachmentToBeReadyTimeout).

Parameter:

Typ	Parameter	Bedeutung
BSTR	fileName	Der Dateiname (inkl. Pfad) der anzuhängenden Datei.

AddBCCRecipient

Fügt der eMail einen neuen BCC-Empfänger hinzu.

Parameter:

Typ	Parameter	Bedeutung
BSTR	mailAddress	eMail-Adresse des BCC-Empfängers.

AddCCRecipient

Fügt der eMail einen neuen CC-Empfänger hinzu.

Parameter:

Typ	Parameter	Bedeutung
BSTR	mailAddress	eMail-Adresse des CC-Empfängers.

AppendToMailBody

Hängt den gegebenen Text an den aktuellen eMail-Text an. Dies funktioniert nur, wenn keine Vorlagendatei für den eMail-Inhalt verwendet wird.

Parameter:

Typ	Parameter	Bedeutung
BSTR	text	Der Text, der an den eMail-Text angehängt werden soll.

AddRecipient

Fügt der eMail einen neuen Empfänger hinzu.

Parameter:

Typ	Parameter	Bedeutung
BSTR	mailAddress	eMail-Adresse des Empfängers

AddVariable

Fügt eine neue Platzhalter-Ersetzung hinzu. Vor dem Senden der eMail wird der gegebene Platzhalter durch den gegebenen Wert ersetzt.

Parameter:

Typ	Parameter	Bedeutung
BSTR	variable	Der Name des zu ersetzenden Platzhalters
BSTR	value	Der Wert, durch den der Platzhalter ersetzt werden soll.

UrlEncode

Codiert den gegebenen String für die Verwendung als URL (z.B. Ersetzung von Leerzeichen durch „%20“, etc.) und liefert das codierte Ergebnis als BSTR.

Parameter:

Typ	Parameter	Bedeutung
BSTR	source	Der zu codierende Wert.

Eigenschaften

BodyEncoding

Die Codierung des eMail-Inhalts, beispielsweise „utf-8“.

EnableStartTLS

true, wenn STARTTLS Verschlüsselung eingesetzt werden soll.

IsMailBodyHtml

true, wenn der eMail-Inhalt HTML enthält.

MailBody

Der Inhalt der eMail. Falls der eMail-Inhalt aus einer Vorlagendatei eingelesen wird (siehe MailTemplateFile) wird diese Eigenschaft ignoriert!

MailTemplateFile

Pfad und Dateiname einer Vorlagendatei, deren Inhalt anstatt MailBody verwendet wird.

RequestError

Liefert true, falls ein Fehler aufgetreten ist, sonst false.

RequestErrorMessage

Liefert, falls RequestError den Wert true hat, eine Fehlermeldung, welche dem Benutzer angezeigt werden kann.

SenderMailAddress

Die eMail-Adresse des Absenders.

SenderName

Der Name des Absenders.

Subject	Der Betreff der eMail.
WaitForAttachmentToBeReady	Gibt an, ob beim Hinzufügen von Anhängen gewartet werden soll, bis die Datei vorhanden ist (z.B. für dynamisch durch externe Prozesse generierte Dateien).
WaitForAttachmentToBeReadyTimeout	Die Zeit in Sekunden, die auf Anhangsdateien gewartet werden soll. Der Wert muss größer als 0 sein.