



# Installation und Benutzung AD.NAV.GZipTools

---

Version 1.0.0.0

**ALTENBRAND Datentechnik GmbH**

Am Gelicht 5

35279 Neustadt (Hessen)

Tel: 06692/202 290

Fax: 06692/204 741

eMail: [support@altenbrand.de](mailto:support@altenbrand.de)

Die Komponente AD.NAV.GZipTools für Microsoft Dynamics NAV ermöglicht es Ihnen, Dateien im GZip-Format zu komprimieren oder GZip-komprimierte Dateien zu dekomprimieren.

## 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

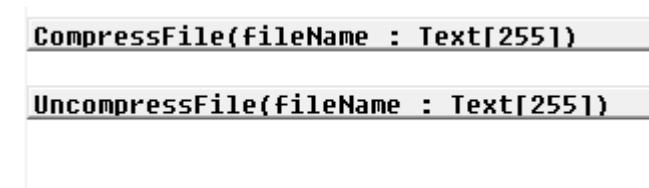
Je nach eingesetzter Version von Microsoft Dynamics NAV gibt es verschiedene Möglichkeiten, die Komponente einzubinden.

### 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.

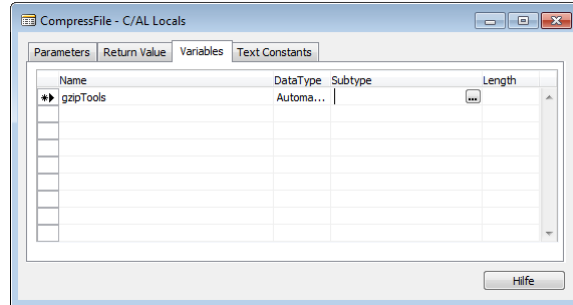
Im folgenden Beispiel werden Functions für das Komprimieren und Dekomprimieren von Dateien erstellt.

Erzeugen Sie eine neue CodeUnit und legen Sie in dieser über „Ansicht > Globals“ zwei neue Functions an, welche jeweils einen Dateinamen als Parameter annehmen:

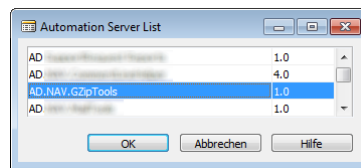


Legen Sie nun in jeder der beiden Functions zunächst mit „Ansicht > Locals“ zwei neue lokale Variablen für den Zugriff auf die Komponente an.

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

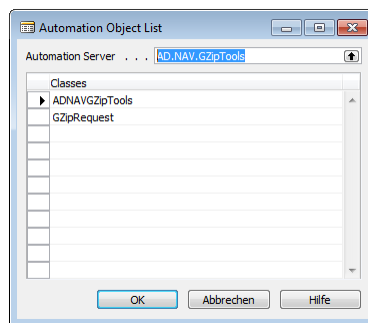


Wechseln Sie in das Feld SubType und drücken Sie die Taste F6. Nun öffnet sich die Maske „Automation Object List“, welche noch leer ist, da zunächst ein Automation Server gewählt werden muss. Drücken Sie hierzu wiederum F6 und öffnen Sie so die Maske „Automation Server List“. In dieser Maske werden alle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



Suchen Sie die den Eintrag `AD.NAV.GZipTools`, 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 `ADNavGZipTools`.



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

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

**CompressFile(fileName : Text[255])**

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

// Erzeuge Automatisierungsobjekte (Role Tailored Client)
//CREATE(gzipTools, TRUE, TRUE);
//CREATE(request, TRUE, TRUE);

// Initialisiere den Request
request.SourceFile := fileName;
request.DestinationFile := fileName + '.gz';

// Datei komprimieren (Bei Fehler wird eine Message ausgegeben)
gzipTools.GZipFile(request);
IF (request.RequestError) THEN
BEGIN
    MESSAGE(request.RequestErrorMessage);
END;

// Automatisierungsobjekte freigeben
CLEAR(request);
CLEAR(gzipTools);
```

**UncompressFile(fileName : Text[255])**

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

// Erzeuge Automatisierungsobjekte (Role Tailored Client)
//CREATE(gzipTools, TRUE, TRUE);
//CREATE(request, TRUE, TRUE);

// Initialisiere den Request (Ausgabedateiname ist Dateiname ohne .gz)
request.SourceFile := fileName;
request.DestinationFile := COPYSTR(fileName, 1, STRLEN(fileName)-3);

// Datei dekomprimieren (Bei Fehler wird eine Message ausgegeben)
gzipTools.GUnzipFile(request);
IF (request.RequestError) THEN
BEGIN
    MESSAGE(request.RequestErrorMessage);
END;

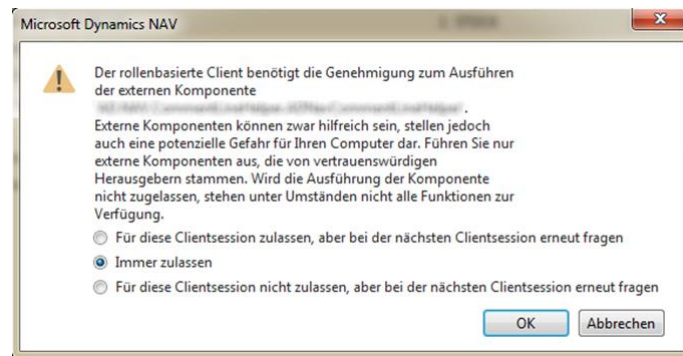
// Automatisierungsobjekte freigeben
CLEAR(request);
CLEAR(gzipTools);
```

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 auf 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 ggf. 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).

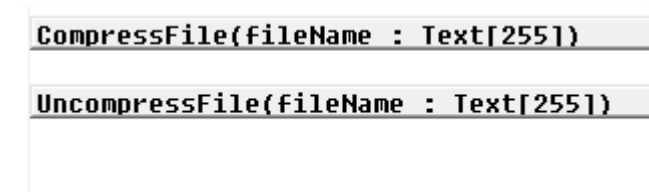


## .NET DLL (Datentyp DOTNET) ab NAV 2013

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

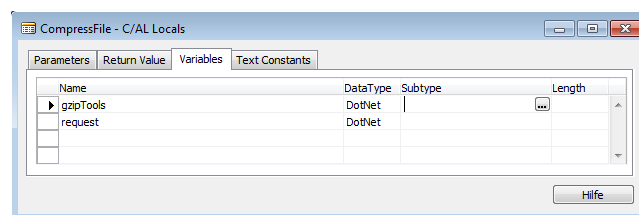
Im folgenden Beispiel werden Functions für das Komprimieren und Dekomprimieren von Dateien erstellt.

Erzeugen Sie eine neue CodeUnit und legen Sie in dieser über „Ansicht > Globals“ zwei neue Functions an, welche jeweils einen Dateinamen als Parameter annehmen:

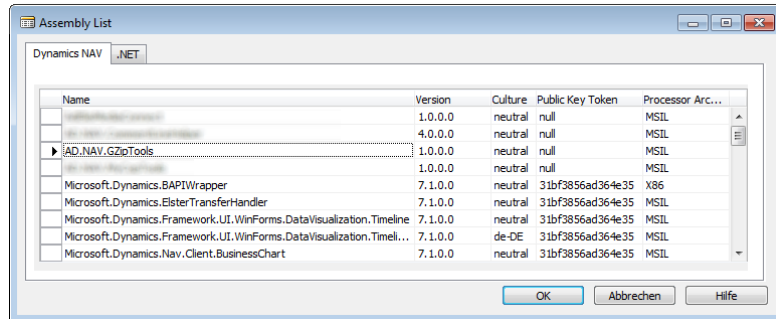


Legen Sie nun in jeder der beiden Functions zunächst mit „Ansicht > Locals“ zwei neue lokale Variablen für den Zugriff auf die Komponente an.

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

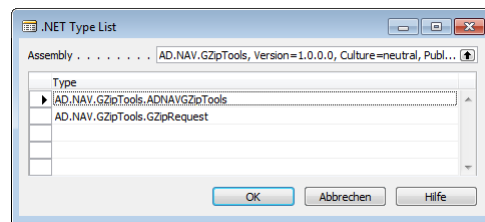


Wechseln Sie in das Feld SubType und drücken Sie die Taste F6. Nun öffnet sich die Maske „.NET Type List“, welche noch leer ist, da zunächst eine Assembly gewählt werden muss. Drücken Sie hierzu wiederum F6 und öffnen Sie so die Maske „Assembly List“. In dieser Maske werden die registrierten Assemblies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

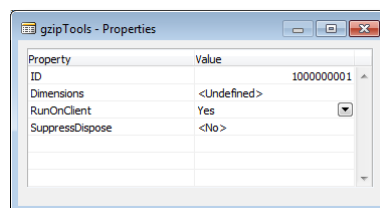


Suchen Sie im Reiter „Dynamics NAV“ den Eintrag AD.NAV.GZipTools, markieren Sie ihn mit der Maus und drücken Sie dann OK.

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



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



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

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

```
CompressFile(fileName : Text[255])
// Erzeuge .NET-Objekte
gzipTools := gzipTools.ADNVAVGZipTools();
request := request.GZipRequest();

// Initialisiere den Request
```

```

request.SourceFile := fileName;
request.DestinationFile := fileName + '.gz';

// Datei komprimieren (Bei Fehler wird eine Message ausgegeben)
gzipTools.GZipFile(request);
IF (request.RequestError) THEN
BEGIN
    MESSAGE(request.RequestErrorMessage);
END;

// .NET-Objekte freigeben
CLEAR(request);
CLEAR(gzipTools);

UncompressFile(fileName : Text[255])
// Erzeuge .NET-Objekte
gzipTools := gzipTools.ADNAVGZipTools();
request := request.GZipRequest();

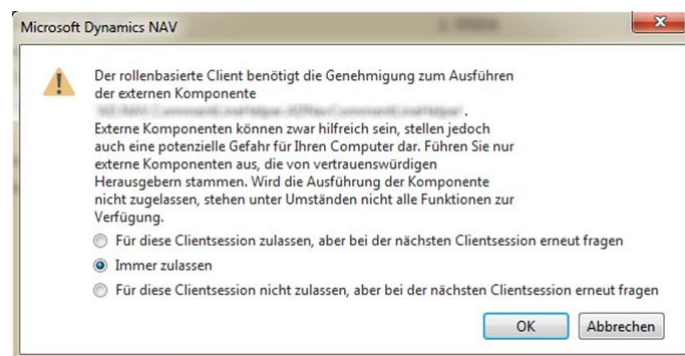
// Initialisiere den Request (Ausgabedateiname ist Dateiname ohne .gz)
request.SourceFile := fileName;
request.DestinationFile := COPYSTR(fileName, 1, STRLEN(fileName)-3);

// Datei dekomprimieren (Bei Fehler wird eine Message ausgegeben)
gzipTools.GUnzipFile(request);
IF (request.RequestError) THEN
BEGIN
    MESSAGE(request.RequestErrorMessage);
END;

// .NET-Objekte freigeben
CLEAR(request);
CLEAR(gzipTools);

```

Bei der ersten Verwendung der Funktionen müssen Sie ggf. 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 generelles Konzept für die Nutzung dieser Komponente ist, dass alle 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
GZipFile	GZipRequest	Komprimiert eine Datei in das GZip-Format.
GUnzipFile	GZipRequest	Dekomprimiert eine Datei aus dem GZip-Format.

## RequestObjects für die Funktionsaufrufe

### GZipRequest

Dieses RequestObject wird für die Funktionen GZipFile und GUnzipFile verwendet. Es ermöglicht die Übergabe von Quell- und Zieldateinamen für den (De-)Komprimierungsvorgang.

### Eigenschaften

SourceFile	Der Name der Quelldatei. Beim Komprimieren ist dies der Name der zu komprimierenden Datei. Beim Dekomprimieren ist dies der Name der Datei im GZip-Format.
DestinationFile	Der Name der Zieldatei. Beim Komprimieren ist dies der Name der zu erzeugenden Datei im GZip-Format. Beim Dekomprimieren ist dies der Name der entpackten Datei.
RequestError	Liefert <code>true</code> , falls ein Fehler aufgetreten ist, sonst <code>false</code> .
RequestErrorMessage	Liefert, falls RequestError den Wert <code>true</code> hat, eine Fehlermeldung, welche dem Benutzer angezeigt werden kann.