



excellent software by

**iXTS**  
excellent software

# **iXTS OPC4ADO**

**Benutzerhandbuch**

# iXTS OPC4ADO Benutzerhandbuch

---

**Dieses Handbuch ist ein Produkt der iXTS GmbH, Germany**

iXTS GmbH  
Rotwandstraße 12  
85609 Aschheim  
Germany  
Tel: +49 89 95 84 08 - 0  
Fax: +49 89 95 84 08 -19  
<http://www.iXTS.de>

Die iXTS GmbH übernimmt für diese Publikation weder implizit noch explizit Haftung irgendwelcher Art. Diese Haftungseinschränkung schließt auch, jedoch nicht ausschließlich, die wirtschaftliche Verwertbarkeit dieses Produkts und seine Tauglichkeit für bestimmte Zwecke ein. Der Benutzer trägt sämtliche Risiken, die aus der Verwendung dieser Information resultieren.

In keinem Fall kann die iXTS GmbH für mittelbare und unmittelbare, zufällige oder besondere Schäden oder Folgeschäden, die aus einem Mangel dieser Dokumentation resultieren, haftbar gemacht werden. Dies gilt auch, wenn die iXTS GmbH auf die Möglichkeit eines solchen Schadens hingewiesen wurde.

Darüber hinaus behält sich die iXTS GmbH das Recht vor, diese Publikation zu bearbeiten und gelegentlich zu verändern, ohne verpflichtet zu sein, irgendeine Person oder Organisation über solch eine Überarbeitung oder Änderung zu unterrichten.

---

## Warenzeichen

Alle Markenzeichen und Produktnamen, die in diesem Handbuch erwähnt werden, hier aber nicht aufgeführt sind, sind Warenzeichen oder eingetragene Warenzeichen ihrer jeweiligen Eigentümer.

## Technische Unterstützung

Falls Sie weitere Fragen zum Einsatz dieses Produkts haben, wenden Sie sich bitte per E-Mail an:

**support@iXTS.de**

## Vertraulichkeit

Die in diesem Dokument veröffentlichten Informationen sind vertraulich und urheberrechtlich geschützt. Sie dürfen ohne vorheriges schriftliches Einverständnis der iXTS GmbH weder direkt noch indirekt veröffentlicht oder in sonstiger Form Dritten zugänglich gemacht werden.

Dieses Dokument darf ohne vorheriges, schriftliches Einverständnis der iXTS GmbH weder im ganzen, noch in Teilen, in irgendeiner Form oder durch den Einsatz von irgendwelchen Hilfsmitteln reproduziert, in einem Datenabfragesystem gespeichert, übersetzt oder übertragen werden.

**© Copyright 2008, iXTS GmbH, Germany. Alle Rechte vorbehalten.**

# Inhaltsverzeichnis

<b>Einleitung</b>	<b>2</b>
Was ist der OPC Server for ADO (OPC4ADO) .....	2
Wer sollte dieses Handbuch lesen.....	2
Aufbau dieses Handbuchs .....	2
Trial Version .....	3
OPC Standardkonformität.....	3
Weitere Quellen .....	3
<b>Inbetriebnahme</b>	<b>4</b>
Systemvoraussetzungen.....	4
Softwarevoraussetzungen .....	4
Hardwarevoraussetzungen .....	4
Installation .....	4
Lizenzierung und Aktivierung.....	4
Lizenzierung.....	5
Aktivierung .....	6
<b>Systembeschreibung</b>	<b>7</b>
Überblick .....	7
Wann wird OPC4ADO benötigt.....	7
Die Technik dahinter .....	8
Datencache und lokale Pufferung.....	9
Registrierung und Deregistrierung einer Installation .....	9
Parallele Installation mehrerer Server .....	10
Datenbankschnittstelle und -kopplung.....	10
Format der Namespace SQL-Query.....	11
Format der Property SQL-Query.....	13
Umsetzung der Signalgüte (Signal-Quality).....	14
Implizite Zuweisung der Signal-Qualities.....	14
Signalgüte-Optionen .....	15
Statistiken und Lebenszeichen .....	15
Elementfunktionen .....	16
Verfügbare Funktionen .....	16
Vorgehen beim Deklarieren einer Elementfunktion .....	17
Dateien.....	17
Programmoptionen (.ini) .....	18
Konfigurationsdatei (.o4a).....	20
<b>Benutzerschnittstelle</b>	<b>24</b>
Das Hauptfenster .....	24
Serverkonfiguration .....	24
Seite Datenbankverbindung.....	25
Seite OPC-Namespace.....	26
Seite OPC-Item-Properties .....	26
Seite OPC-Statistik .....	27
Seite Elementfunktionen .....	29
Seite Signalgüte.....	29
Globale Programmeinstellungen.....	30

Seite Allgemein .....	31
Seite Servereinstellungen .....	31
Seite Ereignisprotokoll .....	32
<b>Diagnose</b>	<b>35</b>
Ereignisprotokoll.....	35
<b>Index</b>	<b>36</b>

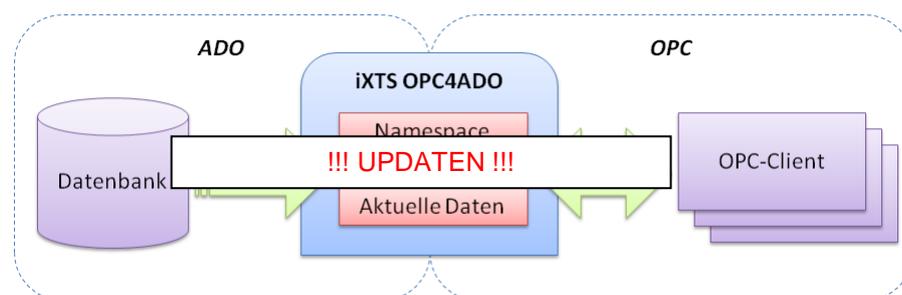
# **iXTS OPC Server for ADO (OPC4ADO) Benutzer- handbuch**

# Einleitung

## Was ist der OPC Server for ADO (OPC4ADO)

Der *OPC Server for ADO* (OPC4ADO) ist eine Applikation, die es ermöglicht beliebige Daten aus einer (ADO-)Datenbank über die standardisierte OPC-Schnittstelle (OPC DA 2.05) zur Verfügung zu stellen. Dabei agiert OPC4ADO auf der einen Seite als Datenbank-Client, auf der anderen als OPC-DA-Server.

Kernkonzept ist die Projektion relationalen Datenstrukturen in den hierarchisch organisierten OPC-Namensraum. OPC4ADO übernimmt das zyklische Auslesen der in der Datenbank persistierten Werte und leitet diese wiederum an die verbundenen OPC-Clients weiter.



Größter Vorteil von OPC4ADO ist die frei konfigurierbare Kopplung zwischen beliebigen Datenbanken und der OPC-Schnittstelle. Damit steht Ihnen ein breites Anwendungsgebiet mit vielerlei Einsatzmöglichkeiten zur Verfügung.

## Wer sollte dieses Handbuch lesen

Dieses Handbuch richtet sich an alle Benutzer von OPC4ADO. Es behandelt die Installation, Inbetriebnahme und Konfiguration der Software und erläutert einige grundsätzliche Vorgehensweisen.

## Aufbau dieses Handbuchs

Dieses Handbuch besteht aus mehreren Abschnitten und ist wie folgt aufgebaut:

**Einleitung:** Dieses einleitende Kapitel

**Inbetriebnahme:** Eine Kurzanleitung zur Erstinbetriebnahme des Systems

**Systembeschreibung:** Beschreibung der in OPC4ADO angewandten Konzepte

**Benutzerschnittstelle:** Eine detaillierte Beschreibung der Benutzerschnittstelle, sowie allen Dialogen

**Diagnose:** Hinweise zur Verwendung der Protokollierungsmöglichkeiten der Applikation

---

## Trial Version

Beim ersten Start von OPC4ADO werden Sie aufgefordert die Applikation zu lizenzieren. Besitzen Sie noch keine gültige Lizenz, können Sie OPC4ADO im vollen Funktionsumfang **für jeweils eine Stunde pro Sitzung testen**. Nach Ablauf einer Stunde OPC4ADO automatisch und ohne Vorwarnung. Sie können jedoch die Applikation erneut starten um weiter zu testen.

Wenn Sie bereits im Besitz einer Lizenz oder eines Lizenzschlüssels sind, können Sie OPC4ADO beim Start, oder auch während der 60-Minuten-Testphase über das Menü „Hilfe/Info...“ lizenzieren.

Erfahren Sie mehr über das Thema im Kapitel [Lizenzierung und Aktivierung](#).

---

## OPC Standardkonformität

OPC4ADO ist konform zur OPC DA Spezifikation in der Version 2.05. Es unterstützt alle vorgeschriebenen Interfaces und zusätzlich das optionale Interface IOPCBrowserServerAddressSpace. Public groups werden nicht unterstützt, ebenso wenig die überholten Interfaces IOPCAsyncIO und IDataObject, da diese eigentlich nur mehr ein Relikt aus der Version 1.0 der Spezifikation sind und normalerweise von OPC DA 2.05 Servers und Clients nicht mehr verwendet werden.

Die Standardkonformität wurde mit dem offiziellen OPC Compliance Test Tool in der Version 2.10 der OPC Foundation getestet und nachgewiesen.

---

## Weitere Quellen

Weitere Informationen zum Thema OPC finden Sie insbesondere auf der Website der OPC Foundation unter <http://www.opcfoundation.org>.

OPC4ADO basiert auf der OPC Data Access Custom Interface Specification Version 2.05, die ebenfalls auf der Seite der OPC Foundation erhältlich ist.

# Inbetriebnahme

---

## Systemvoraussetzungen

### Softwarevoraussetzungen

OPC4ADO setzt mindestens folgende Softwareumgebung voraus:

- Microsoft Windows 2000/XP/Vista/Server 2003
- GDI-Plus bei Windows 2000 (gdiplus.dll)
- Internet Explorer 4.01 Service Pack 1 oder höher
- OPC Core Components 2.00 oder höher

### Hardwarevoraussetzungen

OPC4ADO setzt mindestens folgende Hardwareumgebung voraus:

- Intel Pentium II 500 MHz
- 128 MB RAM
- 20 MB freier Speicherplatz auf der Festplatte

---

## Installation

Wenn Sie sicher sind, dass Ihr System den [Mindestanforderungen](#) genügt, können Sie OPC4ADO sofort in Betrieb nehmen.

OPC4ADO kommt ohne Installationsroutine aus und kann durch einfaches Entpacken des ZIP-Archivs in einen beliebigen Ordner installiert werden. Sollten Sie bereits im Besitz einer Lizenzdatei sein (.license), benennen Sie diese – falls nötig – in **iXTSOPC4ADO.license** um legen Sie sie im selben Verzeichnis ab. Anderenfalls folgen Sie den Anweisungen am Bildschirm während des ersten Starts der Applikation, oder erfahren Sie mehr zu diesem Thema im Kapitel [Lizenzierung und Aktivierung](#).

Damit die neue Installation als OPC-DA-Server von Clientanwendungen gefunden werden kann, muss diese entsprechend am System registriert werden. Führen Sie hierzu einfach iXTSOPC4ADO.exe einmal aus und OPC4ADO erledigt sämtliche Registrierungen selbstständig.

Möchten Sie OPC4ADO mehrmals auf einem System installieren, oder interessieren Sie sich genauer für das Thema der Serverregistrierung, dann können Sie mehr im Kapitel DCOM-Registrierung erfahren.

---

## Lizenzierung und Aktivierung

Wenn Sie OPC4ADO noch nicht lizenziert oder aktiviert haben, werden Sie bei jedem Programmstart dazu aufgefordert (Ausnahme OPC4ADO wurde aufgrund einer Anfrage eines OPC-Clients über DCOM gestartet).

Das folgende Kapitel beschreibt das Vorgehen, um OPC4ADO korrekt zu lizenzieren und zu aktivieren.

Die Lizenzierung/Aktivierung ist auch während der Programmausführung über den Befehl *Hilfe/Info* durchführbar. Öffnen Sie hierzu den Info-Dialog und klicken Sie auf die Schaltflächen **Jetzt lizenzieren** oder **Jetzt aktivieren**.

Ist Ihre Kopie von OPC4ADO nicht lizenziert/aktiviert, läuft die Applikation als [Trial Version](#).\_D2HLink\_2625

Möchten Sie eine Lizenz für OPC4ADO erwerben, haben Sie Fragen zur, oder Probleme während der Lizenzierung und Aktivierung, dann wenden Sie sich bitte an uns:

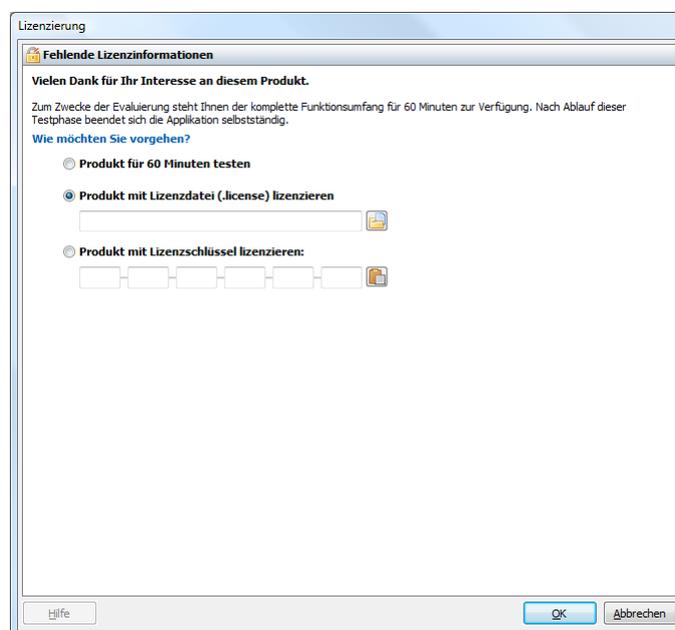
**iXTS GmbH**  
Rotwandstraße 12  
D-85609 Aschheim  
Tel.: +49 89 95 84 08 -0  
Fax.: +49 89 95 84 08 -19  
E-Mail: [licensing@ixts.de](mailto:licensing@ixts.de)  
Web: <http://www.ixts.de>

## Lizenzierung

Die Lizenzierung ist erforderlich, um sicherzustellen, dass es sich bei Ihrer Kopie von OPC4ADO um ein ordnungsgemäß erworbenes Abbild der Software handelt. Die Lizenzierung kann entweder mit einem Lizenzschlüssel (xxxxx-xxxxx-xxxxx-xxxxx-xxxxx-xxxxx), oder einer Lizenzdatei (mit Endung *.license*) durchgeführt werden. Die Lizenzierung ist zum Programmstart, oder während der Programmausführung über den Befehl *Hilfe/Info* durchführbar.

Um OPC4ADO zu lizenzieren, wählen Sie eine der entsprechenden Optionen. Bei der Verwendung der Lizenzdatei, wird die ausgewählte Datei in das Installationsverzeichnis kopiert und in *iXTSOPC4ADO.license* umbenannt.

Sollte während der Lizenzierung ein Fehler auftreten, erscheint dieser am oberen Rand des Lizenzierungsfensters.



## Aktivierung

Die Aktivierung ist erforderlich, um die Lizenz mit dem System, auf dem OPC4ADO installiert wurde, zu assoziieren. Die Aktivierung ist zum Programmstart, oder während der Programmausführung über den Befehl *Hilfe/Info* durchführbar.

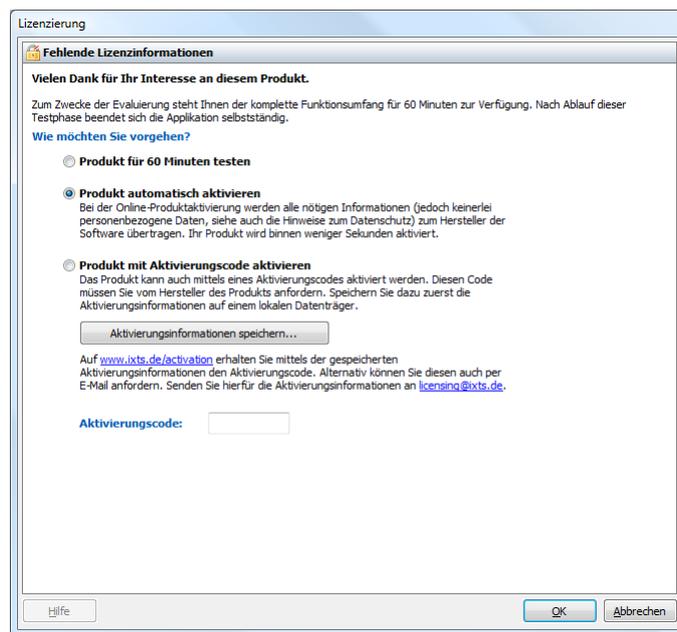
Um OPC4ADO zu aktivieren, wählen Sie eine der entsprechenden Optionen:

Verfügt Ihr System über eine Internetanbindung, wählen Sie die Option **Produkt automatisch aktivieren**. Sollte in Ihrem Netz ein Proxy dazwischengeschaltet sein Übernimmt die automatische Aktivierung die **Proxy-Einstellungen des Microsoft Internet Explorers**.

Sollte Ihr System über keine Internetanbindung verfügen, können Sie Ihr Produkt auch mithilfe eines **Aktivierungscodes** aktivieren. Speichern Sie hierfür zunächst eine Datei mit den für die Aktivierung erforderlichen **Aktivierungsinformationen** (*ActivationRequest.xml*). Die Datei mit den Aktivierungsinformationen können Sie im zweiten Schritt über die Web-Seite <http://www.ixts.de/activation> zur schnellen Generierung Ihres Aktivierungscodes verwenden, oder Sie schicken die Aktivierungsinformationen per E-Mail an [licensing@ixts.de](mailto:licensing@ixts.de). Innerhalb eines Werktages erhalten Sie Ihren Aktivierungscode in einer Antwort-E-Mail. Durch Eingabe des Aktivierungscodes wird OPC4ADO aktiviert.

**Hinweis:** Die Generierung des Aktivierungscodes über die Web-Seite kann von jedem beliebigen System aus durchgeführt werden.

Sollte während der Aktivierung ein Fehler auftreten, erscheint dieser am oberen Rand des Aktivierungsfensters.

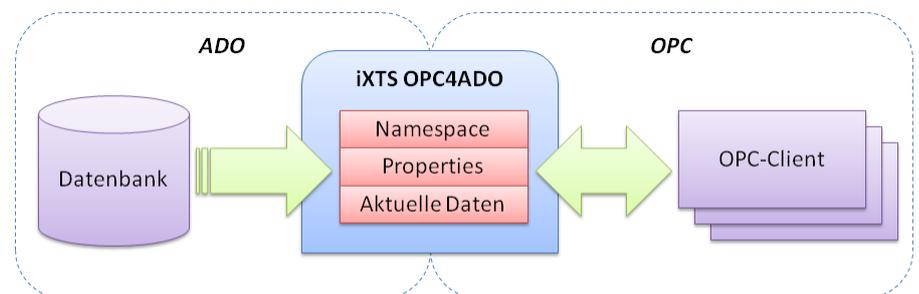


# Systembeschreibung

## Überblick

OPC4ADO („*OPC for ADO*“) ist eine Applikation, die es ermöglicht beliebige Daten aus einer (ADO-) Datenbank über die standardisierte OPC-Schnittstelle (OPC-DA 2.05) zur Verfügung zu stellen. Dabei agiert OPC4ADO auf der einen Seite als Datenbank-Client, auf der anderen als OPC-DA-Server.

Kernkonzept ist die Projektion relationalen Datenstrukturen in den hierarchisch organisierten OPC-Namensraum. OPC4ADO übernimmt das zyklische Auslesen der in der Datenbank persistierten Werte und leitet diese wiederum an die verbundenen OPC-Clients weiter.

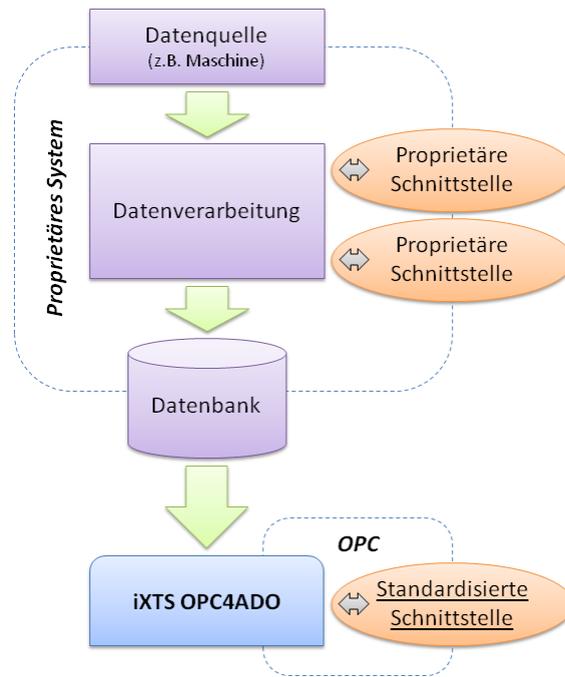


Größter Vorteil von OPC4ADO ist die frei konfigurierbare Kopplung zwischen beliebigen Datenbanken und der OPC-Schnittstelle. Damit steht Ihnen ein breites Anwendungsgebiet mit vielerlei Einsatzmöglichkeiten zur Verfügung.

## Wann wird OPC4ADO benötigt

Immer wenn ein Drittanbietersystem den Zugriff auf die ermittelten Betriebsdaten über OPC nicht implementiert, kommt OPC4ADO zum Einsatz. Einzige Voraussetzung: Die Betriebsparameter müssen in irgendeiner Form in einer Datenbank gespeichert sein.

Folgendes Beispiel soll eine mögliche Anwendung von OPC4ADO veranschaulichen:



Dieses Beispiel zeigt ein gängiges Anwendungsszenario in der Zustandüberwachung von industriellen Maschinen: Ein proprietäres System ist für die Erfassung der Betriebs- und Zustandsdaten von Maschinen zuständig und stellt diese meist über binäre, herstellereigenspezifische und undokumentierte Kommunikationsschnittstellen anderen Anwendungen zur Verfügung. Ist das Datenmodell der Datenbank bekannt, kann OPC4ADO als Mittler zwischen der Drittanbieteranwendung und dem OPC-Universum eingesetzt werden.

## Die Technik dahinter

Generell lässt sich das Laufzeitverhalten von OPC4ADO in zwei voneinander getrennte Phasen einteilen:

- Initialisierung
- Datenerfassung

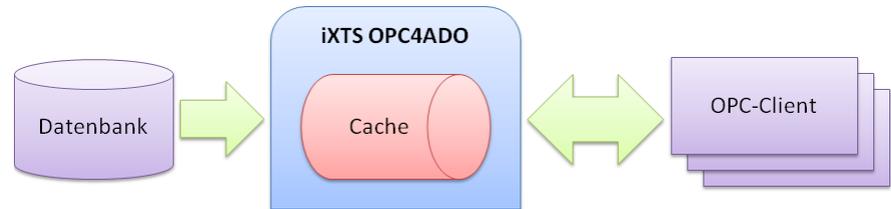
In der Initialisierungsphase wird jedes in der Datenbank gespeicherte Datum als eigenes Element in den hierarchisch organisierten OPC-Namensraum überführt. Zusätzlich speichert OPC4ADO Informationen zu jedem Element ab, wo (Tabelle und Spalte) das entsprechende Datum ausgelesen werden kann. Da diese Kopplung individuell an die zugrundeliegende Datenbasis angepasst werden muss, muss eine [Abfrage](#) in der Abfragesprache **SQL** angegeben werden, die nach Ausführung die für die konkrete Kopplung benötigten Informationen beinhaltet.

Nach Abschluss der Initialisierungsphase befindet sich OPC4ADO in der kontinuierlichen Datenerfassung. Während der Datenerfassung werden zyklisch alle in der Datenbank gespeicherten Daten gelesen und an die angeschlossenen OPC-Clients weitergereicht. Das Aktualisierungsintervall, der Takt mit dem die Datenbank zyklisch ausgelesen wird, kann individuell bestimmt werden.

---

## Datencache und lokale Pufferung

OPC4ADO speichert zunächst alle aus der Datenbank erfassten Daten in einem zentralen Cache, bevor diese im zweiten Schritt an die angeschlossenen OPC-Clients unter Berücksichtigung deren Aktualisierungsintervalle weitergeleitet werden:



Dies entkoppelt die angeschlossenen OPC-Clients von der Datenbank, sodass sogar bei einem plötzlichen Ausfall des Datenbanksservers die Verbindung von OPC-Client zu OPC4ADO aufrecht gehalten werden kann und der zuletzt bekannte Zustand stets abrufbar bleibt.

Um Daten mit hohen Erfassungsraten verlustfrei an Clients mit niedrigeren Erfassungsraten, oder über kurzzeitig ausgelastete Netzwerke übertragen zu können, bietet OPC4ADO die Option der **lokalen Datenpufferung**. Ist die [lokale Datenpufferung aktiviert](#), werden alle bei OPC4ADO eintreffenden Daten in einem FIFO-Puffer (**F**irst **I**n **F**irst **O**ut) gespeichert. OPC4ADO merkt sich für jeden angeschlossenen OPC-Client die zuletzt übermittelten Daten und sendet nur die neu hinzugekommenen Daten mit dem nächsten *Daten-Callback*.

**Hinweis:** Ist die lokale Datenpufferung aktiviert, können *Daten-Callbacks* mehr als ein Datum pro Element beinhalten. Einige OPC DA 2.05 konforme Clients sind nicht in der Lage, diese *Daten-Callbacks* richtig zu verarbeiten. Sollten Sie ein Fehlverhalten feststellen, [ändern Sie die Weiterleitungsstrategie](#) auf **Nur aktuellstes Datum**. Beachten Sie, dass es bei dieser Option wieder zu Datenverlust kommen kann, da nicht alle Daten an die Clients weitergereicht werden.

Wie Sie die lokalen Datenpuffer konfigurieren erfahren Sie im Kapitel [Seite Servereinstellungen](#).

## Registrierung und Deregistrierung einer Installation

Während OPC4ADO startet, wird überprüft, ob die entsprechende Installation bereits registriert ist. Ist dies nicht der Fall, wird automatisch eine Registrierung in der DCOM-Umgebung initiiert.

### Manuelle Registrierung

Möchten Sie eine Registrierung initiieren ohne OPC4ADO zu starten, rufen Sie die entsprechende Installation mit den Kommandozeilenparameter `/REGSERVER` auf (z.B. `C:\iXTSOPC4ADO.exe /UNREGSERVER`).

### Deregistrierung

Bevor Sie OPC4ADO von Ihrem System entfernen, sollten Sie die Installation wieder deregistrieren. Dies bewerkstelligen Sie, indem Sie die entsprechende Installation mit den Kommandozeilenparameter `/UNREGSERVER` aufrufen (z.B. `C:\iXTSOPC4ADO.exe /UNREGSERVER`).

### Verschieben der Installation

Möchten Sie das Verzeichnis einer bereits registrierten Installation ändern, können Sie dies ohne weitere Beachtung vornehmen. Nachdem Sie die Installation verschoben haben, müssen Sie allerdings OPC4ADO einmalig ausführen, damit der Installationspfad entsprechend aktualisiert werden kann. In diesem Anwendungsfall initiiert OPC4ADO eine Deregistrierung der alten Installation, sowie eine Neuregistrierung der neuen Installation mit denselben Angaben.

## Parallele Installation mehrerer Server

OPC4ADO unterstützt mehrere Mechanismen um eine Ausführung mehrerer Installationen auf einem System parallel zu ermöglichen. Zunächst wird eine „Installation“ als eine eigenständige EXE-Datei in einem eigenen Verzeichnis definiert.

Folgende Anwendungsfälle werden von OPC4ADO bezüglich dieser Funktion abgedeckt:

- Registrierung neuer Installationen
- Deregistrierung nicht mehr vorhandener Installation
- Wenn es sich um eine Neuregistrierung handelt und gleichzeitig eine andere Installation nicht mehr vorhanden ist, werden die Einstellungen der nicht mehr vorhandenen Installation übernommen (entspricht „[Verschieben der Installation](#)“)
- Schutz vor Überschreiben der Einstellungen einer Installation, wenn Datei iXTSOPC4ADO.ini kopiert wurde

---

## Datenbankschnittstelle und -kopplung

Dieses Kapitel beschäftigt sich mit dem Thema der Kopplung zwischen Datenbank und OPC4ADO.

OPC4ADO bietet die Funktionalität, jedes in einer Datenbank gespeicherte Modell von Daten über die OPC-DA-Schnittstelle bereitzustellen. Grundvoraussetzung ist, dass das Datenmodell in den OPC-Namensraum überführt werden kann. Hierfür muss jeder in der Datenbank gespeicherte Wert als OPC-Element im Namensraum repräsentiert werden können.

Aufgrund der Tatsache, dass das konkrete Datenmodell der zugrundeliegenden Datenbank individuell ist, kann OPC4ADO nicht über einen allgemeinen Prozess die Kopplung bewerkstelligen. Für die konkrete Anpassung an das zugrundeliegende Datenmodell bietet Ihnen OPC4ADO diverse Konzepte an, primär ist allerdings die sogenannte [Namespace SQL-Query](#) von Bedeutung.

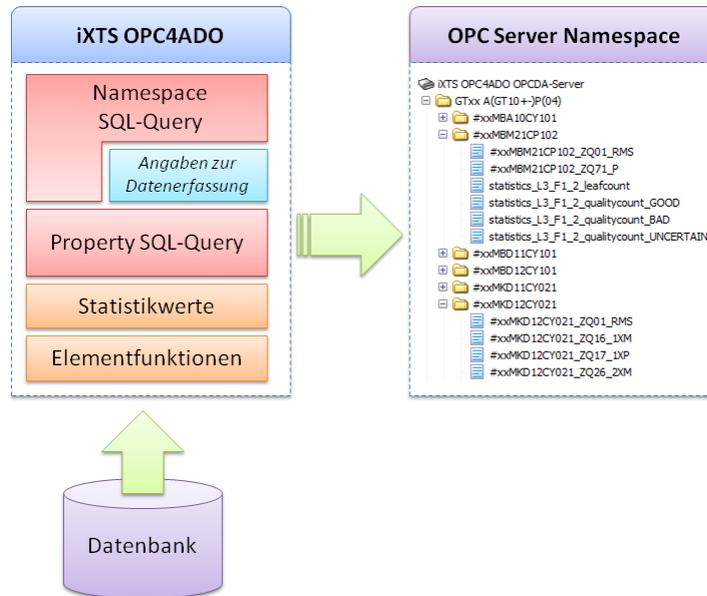
OPC4ADO führt in der [Initialisierungsphase](#) die [Namespace SQL-Query](#) einmalig aus. Das Ergebnis dieser Abfrage erzeugt eine temporäre Tabelle, dessen Spaltennamen einem festen Format gehorchen müssen. OPC4ADO kann anhand der Angaben in den erwarteten Spalten den OPC-Namensraum erzeugen.

Enthält das zugrundeliegende Datenmodell zu jedem Element weitere Informationen, z.B. Extremwerte oder Umgebungsinformationen, können diese als OPC-Properties dem entsprechenden Elementen zugeordnet werden. Die Informationen der OPC-Properties werden in einer separaten SQL-Abfrage, der [Property SQL-Query](#), ebenfalls einmalig während der Initialisierungsphase abgefragt.

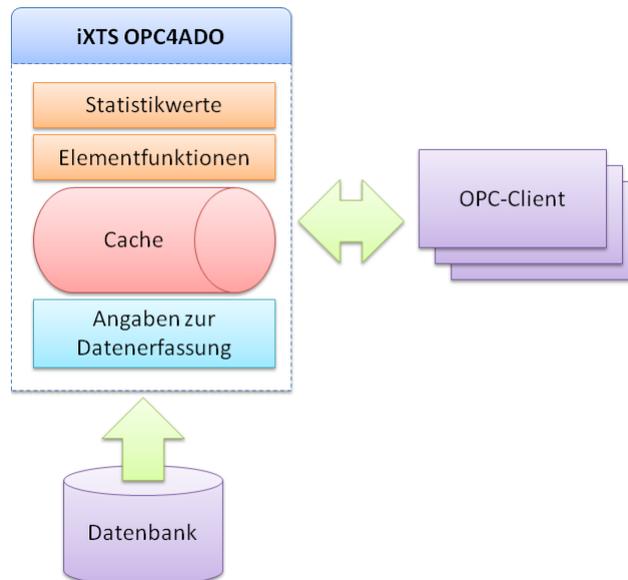
Möchten Sie die aus der Datenbank gewonnenen Daten um weitere Informationen bereichern, stehen Ihnen die Möglichkeiten offen, [Statis-](#)

tiken über alle ermittelten Daten, oder Werte mithilfe gängiger [mathematischer Funktionen](#) zu errechnen und zusätzlich im OPC-Namensraum anzuzeigen.

Folgende Grafik veranschaulicht die für die Generierung des OPC-Namenspaces verwendeten Komponenten:



Neben den für die Generierung des OPC-Namenspaces erforderlichen Angaben enthält die Namespace SQL-Query auch die **Angaben zur Datenerfassung**. Diese legen unter anderem fest, in welcher Tabelle unter welcher Spalte der aktuelle Wert zum entsprechenden Element abgefragt werden kann. Diese Angaben sind für die zyklische Aktualisierung der Daten notwendig. Die folgende Grafik veranschaulicht die während der Datenerfassung beteiligten Komponenten:



## Format der Namespace SQL-Query

Jeder Datensatz (Zeile) der Namespace SQL-Query kann bis zu zwei eigenständige Elemente aus der Datenbank repräsentieren. Sollten Sie von Elementfunktionen Gebrauch machen, sind pro Datensatz zusätzli-

che Angaben erforderlich. Folgende Spalten erwartet OPC4ADO in der durch die Namespace SQL-Query erzeugten Ergebnistabelle:

Spaltenname	Attribute	Beschreibung
<i>Verwaltung</i>		
<b>RecordID</b>	erforderlich, eindeutig	Zeichenfolge zur eindeutigen Identifizierung des aktuellen Datensatzes. Wird benötigt um Bezug zur Property-Query herzustellen.
<b>ItemType</b>	erforderlich	<p><b>Wenn 0</b>, einzelnes Element. Es wurden nur die Angaben zum primären Element verarbeitet. Angaben zum sekundären Element können auf „NULL“ gesetzt werden, da ignoriert.</p> <p><b>Wenn &gt; 0</b>, Element-Paar aus primären und sekundären Wert (primary + secondary). Alle Angaben zum primären, als auch zum sekundären Wert müssen sinnvoll belegt sein.</p> <p>Dient auch zur Separierung des Anwendungsbereichs der Elementfunktionen.</p>
<i>OPC/Namespace</i>		
<b>PrimaryOPCPath</b>	erforderlich, eindeutig	<p>Vollständige Pfadangabe des primären Elements im Namensraum. Ebenen werden durch Backslash \ getrennt. Hinterer Teil entspricht dem Elementnamen.</p> <p>Beispiel: Motherboard\Temperatures\CPUTemp1</p>
<b>PrimaryOPCID</b>	erforderlich, eindeutig	Vollqualifizierte ID des OPC-Elements. Kann z.B. identisch mit Pfad sein.
<b>SecondaryOPCPath</b>	erforderlich wenn ItemType>0, eindeutig	<p>Vollständige Pfadangabe des sekundären Elements im Namensraum. Ebenen werden durch Backslash \ getrennt. Hinterer Teil entspricht dem Elementnamen.</p> <p>Beispiel: Motherboard\Temperatures\CPUTemp1</p>
<b>SecondaryOPCID</b>	erforderlich wenn ItemType>0, eindeutig	Vollqualifizierte ID des OPC-Elements. Kann z.B. identisch mit Pfad sein.
<i>Zugriff auf Daten</i>		

<b>SourceTable</b>	erforderlich	Gemeinsamer Name der Datenbanktabelle in der Wert für primären und ggf. sekundären Wert steht.
<b>TimeStampColumn</b>	optional	Gemeinsamer Name der Spalte innerhalb SourceTable in der Zeitstempel für primären und ggf. sekundären Wert steht.  Wenn nicht angegeben wird stets aktueller Zeitstempel verwendet.
<b>PrimaryValueColumn</b>	erforderlich	Name der Spalte innerhalb SourceTable in der Wert für primären Wert steht.
<b>SecondaryValueColumn</b>	erforderlich wenn ItemType>0	Name der Spalte innerhalb SourceTable in der Wert für sekundären Wert steht.
<i>Zugriff auf Datenqualität</i>		
<b>QualityTable</b>	optional	Gemeinsamer Name der Datenbanktabelle in der Qualität für primären und ggf. sekundären Wert steht.
<b>QualityColumn</b>	erforderlich wenn QualityTable≠NULL	Gemeinsamer Name der Spalte innerhalb der QualityTable in der Qualität für primären und ggf. sekundären Wert steht.
<i>Für jede Elementfunktion mit Namen &lt;Name&gt;</i>		
<b>&lt;Name&gt;OPCPath</b>	erforderlich wenn Elementfunktion auf ItemType angewendet	Vollständige Pfadangabe des Elements der Elementfunktion <Name> im Namensraum. Ebenen werden durch Backslash \ getrennt. Hinterer Teil entspricht dem Elementnamen.  Beispiel: Motherboard\Temperatures\Average
<b>&lt;Name&gt;OPCID</b>	erforderlich wenn Elementfunktion auf ItemType angewendet	Vollqualifizierte ID des OPC-Elements. Kann z.B. identisch mit Pfad sein.

## Format der Property SQL-Query

Jeder Datensatz (Zeile) der Property SQL-Query referenziert einen Datensatz aus der Namespace SQL-Query. Der Bezug wird dabei über den Wert der Spalte **RecordID** hergestellt. Die Anzahl der Spalten variiert je nachdem welche OPC-Properties [definiert](#) wurden. Sollten Sie von [Elementfunktionen](#) Gebrauch machen, sind weitere Spalten erforderlich. Folgende Spalten erwartet OPC4ADO in der durch die Property SQL-Query erzeugten Ergebnistabelle:

Spaltenname	Attribute	Beschreibung
<i>Verwaltung</i>		
<b>RecordID</b>	erforderlich, eindeutig	Zeichenfolge zur eindeutigen Identifizierung des aktuellen Datensatzes. Wird benötigt um Bezug zur Namespace-Query herzustellen.
<i>Für jede definierte OPC-Property mit ID &lt;ID&gt;</i>		
<b>PrimaryValuePropertyID&lt;ID&gt;</b>	erforderlich	Wert für OPC-Property, oder NULL. Wenn NULL, wird Standardwert verwendet.
<b>SecondaryValuePropertyID&lt;ID&gt;</b>	erforderlich wenn Elementpaar (ItemID>0)	Wert für OPC-Property, oder NULL. Wenn NULL, wird Standardwert verwendet.
<i>Für jede Elementfunktion mit Namen &lt;Name&gt; + jeder definierten OPC-Property mit ID &lt;ID&gt;</i>		
<b>&lt;Name&gt;PropertyID&lt;ID&gt;</b>	erforderlich	Wert für OPC-Property, oder NULL. Wenn NULL, wird Standardwert verwendet.

## Umsetzung der Signalgüte (Signal-Quality)

Die OPC-DA Spezifikation bestimmt verschiedene sogenannte Signal-Qualities, um den aktuellen Wert um seine *Qualität* zu ergänzen. Dadurch lässt sich z.B. ausdrücken, dass zwar ein Wert ermittelt werden konnte, dieser aber bereits obsolet ist.

OPC4ADO bietet verschiedene Möglichkeiten, Signal-Qualities umzusetzen. Neben einer impliziten Zuweisung durch OPC4ADO selbst, können Sie zusätzlich obsoletere Werte durch eine entsprechende Quality [kennzeichnen](#) und darüber hinaus auch proprietäre Qualitätsinformationen aus der Datenbank in die von OPC spezifizierten [transformieren](#).

### Implizite Zuweisung der Signal-Qualities

Die implizite Zuweisung der Signal-Quality hat in OPC4ADO die höchste Priorität. Wird ein Fehler festgestellt, der direkt einem konkreten OPC-Element zugeschrieben werden kann, wird die Qualität des entsprechenden Elements auf folgende Signal-Quality gesetzt:

Status	Sub-Status	Hex
Bad	Out of service	0x001C

In diesem Fall besteht ein Problem beim Zugriff auf die Datenbank, da mindestens eines der folgenden Zugriffsattribute fehlerhaft ist:

- Wertetabelle nicht gefunden (SourceTable)
- Wertespalte nicht gefunden (Primary-/SecondaryValueColumn)

- Zeitstempelspalte nicht gefunden (TimeStampColumn)
- Qualitätstabelle nicht gefunden (QualityTable)
- Qualitätsspalte nicht gefunden (QualityColumn)

Eine entsprechende Fehlermeldung kann im Hauptfenster unter „Aktuelle Daten“ in der Spalte „Status“ eingesehen werden.

Sind die Attribute für den Datenbankzugriff (Tabelle, Spalte) eines OPC-Elements korrekt, werden im nächsten Schritt die [Signalgüte-Optionen](#) herangezogen.

## Signalgüte-Optionen

Konnten bei einem OPC-Element keine Fehler bezüglich seiner Konnektivität zur Datenbank festgestellt werden, werden im zweiten Schritt die Signalgüte-Optionen herangezogen. Die Signalgüte-Optionen können in der Serverkonfiguration auf der [Seite Signalgüte](#) angepasst werden.

Die Signalgüte-Optionen bestehen aus zwei Teilfunktionen, der **Signalausfallerkennung** und der **Quality-Konvertierung**.

Die **Signalausfallerkennung** kann aktiviert werden, um anhand des Zeitstempels der OPC-Elemente obsoletere Daten zu erkennen. Dabei muss ein maximal zulässiges Datenalter angegeben werden. Je nachdem, ob der Zeitstempel *aktuell* oder *veraltet* ist, kann eine entsprechende Signal-Quality gesetzt werden.

Sollte die Datenbank selbst Qualitätsangaben beinhalten, welche sich aber nicht ohne weiteres in das von OPC geforderte Format überführen lassen, kann mithilfe der **Quality-Konvertierung** eine Tabelle angelegt werden, die eine genaue Konvertierung vom Datenbank-Format in das OPC-Quality-Format beschreibt. Für dieses Feature müssen in der Namespace SQL-Query entsprechende Informationen zu *QualityTable* und *QualityColumn* vorhanden sein. Sollte die Signalausfallerkennung aktiviert sein, kann die Übersetzung zusätzlich zwischen *aktuellen* und *veralteten* Daten unterscheiden.

Erfahren Sie mehr über Definition und Konfiguration der Signalgüte-Optionen unter [Signalgüte](#).

---

## Statistiken und Lebenszeichen

OPC4ADO bietet Ihnen die Möglichkeit, sogenannte Statistikelemente zusätzlich im OPC-Namensraum erzeugen zu lassen. Konfigurieren Sie die Optionen der Statistikelemente in der Serverkonfiguration auf der [Seite OPC-Statistik](#).

OPC4ADO stellt Ihnen drei Kategorien von Statistikelementen zur Verfügung:

- Zählerelemente
- Statusinformationen
- Lebenszeichen

Zu den Zählerelementen gehören jene Statistikelemente, welche alle OPC-Elemente zählen, zu denen eine bestimmte Bedingung zutrifft. Aktuell können alle Elemente, Elemente mit Signal-Quality GOOD, UNCERTAIN und BAD gezählt werden. **Dabei werden alle Statistikelemente von der Zählung ausgeschlossen.**

Generell werden diese Statistikelemente im Wurzelverzeichnis des OPC-Namensraums erzeugt. Es gibt jedoch die Möglichkeit, die zählenden Statistikelemente in jedem Unterknoten des OPC-Namensraums erzeugen zu lassen. Bei dieser Option werden alle Elemente gezählt, die ab entsprechendem Unterknoten vorhanden sind.

Zu den Statuselementen zählen die Statistiken „Ältestes Datenalter“ und „Zulässiges Datenalter überschritten“. Diese Statistikelemente können im Kontext der Signalausfallerkennung von Interesse sein.

Lebenszeichen signalisieren einem angeschlossenen OPC-Client auf einfache Weise, dass die Verbindung zu OPC4ADO in Ordnung ist. Hierfür stehen Ihnen die Elemente „Heartbeat“ und „Laufzeit des Servers“ zu Verfügung. Während „Heartbeat“ bei null beginnt und sekundlich um Eins erhöht wird, liefert „Laufzeit des Servers“ die vergangene Zeit als Datum/Zeit.

Erfahren Sie mehr über die Konfiguration der Statistikelemente unter [OPC-Statistik](#).

## Elementfunktionen

Nutzen Sie die Elementfunktionen, um Werte aus der Datenbank miteinander zu verrechnen und die neu errechneten Werte als neue Elemente im OPC-Namensraum anzuzeigen. Deklarieren Sie Elementfunktionen in der Serverkonfiguration auf der [Seite Elementfunktionen](#).

Eine Elementfunktion wird immer auf alle [Datensätze der Namespace SQL-Query](#) angewendet, die einen bestimmten Elementtyp ([ItemType](#)) entsprechen. Als Eingabeparameter der gewählten Funktion dienen dabei der primäre und der sekundäre Wert des jeweiligen Datensatzes.

Um die Platzierung der neu gewonnenen Elemente im OPC-Namensraum anzupassen, müssen pro Elementfunktion zwei neue Spalten in der OPC Namespace-Query hinzugefügt werden, um zum einen den Namespace-Pfad, als auch die OPC-ID ermitteln zu können.

Damit die neu gewonnenen Elemente auch mit OPC-Properties behaftet werden können, muss die Property SQL-Query für jede Elementfunktion und für jede definierte OPC-Property mit den entsprechenden Spalten ergänzt werden.

Mehr über das Erweitern der Namespace und Property SQL-Query können Sie den Kapiteln [Format der Namespace SQL-Query](#) und [Format der Property SQL-Query](#) entnehmen.

## Verfügbare Funktionen

Derzeit bietet OPC4ADO folgende Funktionen an:

Term	Parameter	Beschreibung
<b><i>primary</i></b> <b><i>cos(secondary)</i></b> *	<i>Primary:</i> Amplitude/Länge des Polarvektors	Errechnet den X-Anteil eines Polarvektors, dessen Länge durch den primären und dessen Winkel durch den sekundären Wert beschrieben wird.
	<i>Secondary:</i> Phase/Winkel	

	des Polarvektors in Grad {0..360}	
<b>primary</b> <b>sin(secondary)</b>	* Primary: Amplitude/Betrag des Polarvektors Secondary: Phase/Winkel des Polarvektors in Grad {0..360}	Errechnet den Y-Anteil eines Polarvektors, dessen Länge durch den primären und dessen Winkel durch den sekundären Wert beschrieben wird.

## Vorgehen beim Deklarieren einer Elementfunktion

Um eine Elementfunktion zu deklarieren, sollten Sie wie folgt vorgehen:

### 1. Namespace SQL-Query erstellen und testen

- Wertepaare entsprechend zusammenstellen
- Wertepaare entsprechend separieren durch unterschiedliche *ItemTypes*
- Namespace SQL-Query zunächst testen und verifizieren

### 2. Property SQL-Query erstellen und testen

### 3. Elementfunktionen deklarieren

- Funktionen deklarieren
- *ItemTypes* für Anwendung entsprechend setzen
- Zusammenstellen der zusätzlichen Spaltennamen für Namespace und Property SQL-Query

### 4. Namespace SQL-Query um zusätzliche Spalten erweitern

### 5. Property SQL-Query um zusätzliche Spalten erweitern

### 6. Korrekte Funktion verifizieren

---

## Dateien

Während der Verwendung von OPC4ADO können verschiedene Dateien erzeugt werden. Zum einen werden alle Programmoptionen in einer INI-Datei abgelegt, zum anderen kann die konkrete Serverkonfiguration in sogenannten Konfigurationsdateien mit der Dateiendung .o4a abgelegt werden.

Die Separierung der Informationen in INI- und Konfigurationsdatei lässt sich dabei wie folgt vereinheitlichen:

- Alle für den *normalen Betrieb* erforderlichen Angaben, die nicht je unterstützter Datenbank variieren können, befinden sich in der INI-Datei. Diese Einstellungen werden in der Regel über den Dialog „[Globale Programmeinstellungen](#)“ vorgenommen.
- Alle für die konkrete Unterstützung einer Datenbank relevanten Informationen befinden sich in den Konfigurationsdateien. Diese Angaben werden im Dialog „[Server konfigurieren](#)“ bearbeitet.

Generell sollte es zu jeder Installation eine INI-Datei und eine Konfigurationsdatei geben.

## Programmoptionen (.ini)

Das aktuelle Format der INI-Datei ist wie folgt:

Sektion/Attribut	Datentyp/Wert	Beschreibung
<i>[Startup]</i>		
<b>StartupType</b>	Aufzählung {0, 1, 2}	Legt fest, ob beim Start eine Datei geladen werden soll.  0 = Es wird keine Datei geladen sondern eine neue, leere Datei bereitgestellt.  1 = Es wird die zuletzt verwendete Datei geöffnet (siehe Sektion „RecentlyUsedConfigurations“)  2 = Es wird eine bestimmte Datei geöffnet (siehe „SpecificStartupConfiguration“)
<b>SpecificStartupConfiguration</b>	Text/Dateiname	Gibt den Dateinamen der Datei an, die immer beim Start geladen werden soll.  Wird ignoriert wenn StartupType ≠ 2.
<i>[MainWindow]</i>		
<b>Left, Top, Width, Height</b>	Ganzzahl [Pixel]	Bestimmt die Koordinaten/Größe des Hauptfensters.
<b>WindowState</b>	Aufzählung {0, 1, 2}	Legt fest, ob das Hauptfenster maximiert (2), minimiert (1) oder normal (0) angezeigt wird.
<b>CustomizedTitle</b>	0/1	Legt fest, ob der Fenstertitel automatisch (0), oder vom Anwender individuell gesetzt wird (1).
<b>CustomizedTitleNormal, CustomizedTitleError</b>	Text	Legt den individuellen Fenstertitel im Falle eines Fehlers, bzw. im normalen Betrieb fest.  Wird ignoriert wenn CustomizedTitle = 0.
<i>[OPCDA ServerIdentity]</i>		
<b>Name, ID, VendorInfo</b>	Text/ID	Bestimmt die Serveridentifikation der entsprechenden Installation.
<i>[LocalBuffering]</i>		
<b>LocalBufferingEnabled</b>	0/1	Legt fest, ob der interne Cache nur den aktuellsten Wert (0)

		speichert, oder Werte puffert (1).
<b>MaxLocalBufferSize</b>	Ganzzahl	Legt fest, wie viele Werte maximal pro OPC-Element im internen Cache gepuffert werden können.  Wird ignoriert wenn LocalBufferEnabled = 0.
<b>TransitionStrategy</b>	Aufzählung {0, 1}	Legt fest, wie gepufferte Daten im internen Cache an die angeschlossenen Clients weitergereicht werden.  0 = Es wird nur das aktuellste Datum weitergereicht; Damit wird die Pufferung deaktiviert (= Kompatibilitätsmodus für problematische OPC-Clients).  1 = Es werden alle für zwischenzeitlich neu gepufferten Elemente weitergereicht.
<i>[EventLogger]</i>		
<b>AsyncMode</b>	0/1	Legt fest, ob das Schreiben der Log-Datei nebenläufig geschehen soll (1), oder unmittelbar nach Auftreten eines Ereignisses (0).  Sollte stets auf 1 bleiben, da bessere Ausführungsleistungen erzielt werden.
<b>LimitMessageRate</b>	0/1	Legt fest, ob die Ereignisprotokollierung bei vielen unmittelbar hintereinander ausgelösten Ereignissen limitiert (1), oder alle Ereignisse (0) erfasst werden sollen.  Sollte stets auf 1 bleiben, da bessere Ausführungsleistungen erzielt werden.
<b>LogDir</b>	Text/Verzeichnis	Gibt das Verzeichnis an, indem die Log-Dateien gespeichert werden. Angabe kann absolut oder relativ zum Installationsverzeichnis gesetzt werden.
<b>LoggingLevel</b>	Aufzählung {0, 1, 2, 3, 4, 5}	Legt den Detaillierungsgrad der Ereignisprotokollierung fest.  0 = Ereignisprotokollierung ist deaktiviert  1 = Sehr gering  2 = Gering  3 = Normal  4 = Erweitert

		5 = Vollständig
<b>MaxDirSize</b>	Ganzzahl [Bytes]	Gibt die maximale Größe des Log-Verzeichnisses an, bevor ältere Dateien überschrieben werden.
<b>MaxFileSize</b>	Ganzzahl [Bytes]	Gibt die maximale Größe einer Log-Datei an, bevor eine neue Log-Datei angelegt wird.
<b>PutErrorsAndWarningsToWindowsEventLog</b>	0/1	Legt fest, ob Ereignisse vom Typ Warnung und Alarm zusätzlich in das Windows-Eventlog geschrieben werden sollen (1).
<b>SaveLog</b>	0/1	Legt fest, ob Log-Dateien geschrieben werden sollen (1), oder nicht (0).
<i>[RecentlyUsedConfigurations]</i>		
Liste zuletzt verwendeter Konfigurationen		
<b>n</b>	Text/Dateiname	Vollständiger Dateiname einer Konfigurationsdatei.

## Konfigurationsdatei (.o4a)

Das Format der Konfigurationsdatei gleicht primär dem einer INI-Datei, jedoch werden bestimmte Angaben mithilfe von TAB-Zeichen separiert:

<b>Sektion/Attribut</b>	<b>Datentyp/Wert</b>	<b>Beschreibung</b>
<i>[MetaInfo]</i>		
<b>FileVersion</b>	Text/Version	Gibt die Dateiversion an. Ist erforderlich um Datei auszulesen.
<b>Generator</b>	Text	Gibt an, welche Anwendung diese Datei erzeugt hat.
<i>[DatabaseConnection]</i>		
<b>ConnectionString</b>	Text/ADO-Verbindungs-kennung	Gibt die vollständige ADO-Verbindungs-kennung zur Datenbank an.
<b>Username</b>	Text	Benutzername zur Identifikation beim Datenbank-Server.
<b>Password</b>	Text	Passwort zur Identifikation beim Datenbank-Server.
<b>ScanRate</b>	Ganzzahl [Millisekunden]	Gibt das Aktualisierungsintervall an, indem OPC4ADO zyklisch aus der Datenbank liest.
<b>NumberAsDoubles</b>	0/1	Gibt an, ob alle aus der Datenbank gelesenen Zahlenwerte vereinheitlicht in 64-Bit-Fließkommazahl (VT_R8) konvertiert (1), oder die ursprüngli-

		chen Datentypen aus der Datenbank beibehalten werden sollen (0).
<i>[NamespaceQuery]</i>		
Enthält vollständige Namespace SQL-Query		
<i>[NamespaceSettings]</i>		
<b>Name-spaceStructure</b>	Aufzählung {0, 1, 2}	Gibt an, wie die Pfadangaben aus der Namespace SQL-Query verarbeitet werden sollen.  0 = Hierarchisch  1 = <i>Flacher</i> Namensraum – Pfadangaben werden ignoriert – nur Name wird verwendet.  2 = <i>Flacher</i> Namensraum mit kompletter Pfadabgabe – Backslashes werden durch Unterstrich ersetzt.
<i>[Statistics.ElementCounter]</i>		
<b>Visible</b>	0/1	Legt fest, ob Zählerelemente im Namensraum erzeugt werden sollen (1), oder nicht (0).
<b>Prefix</b>	Text	Legt das Präfix bei der Erzeugung der Namen fest.
<b>CreateAnonymousNames</b>	0/1	Legt fest, ob bei der Erzeugung der Namen <i>anonyme Namen</i> (1), oder <i>sprechende Namen</i> (0) verwendet werden sollen.
<b>ShowSubLevelItems</b>	0/1	Legt fest, ob die Zählerelemente auf allen Hierarchieebenen (1), oder nur auf oberster Ebene (0) erzeugt werden sollen.
<b>MaxDepth</b>	Ganzzahl	Legt fest, bis zu welcher Hierarchieebene Elemente erzeugt werden sollen.  Wird ignoriert, wenn ShowSubLevelStatistics = 0.
<i>[Statistics.Elements]</i>		
<b>Element (TAB) Visible (TAB) Path</b>		
- <b>Element</b>	Text/intern	Interne ID zur Erkennung des Statistikelements.
- <b>Visible</b>	0/1	Gibt an, ob das Statistikelement angezeigt werden soll (1) oder nicht (0).
- <b>Path</b>	Text/Pfad	Gibt den vollständigen Pfad des Elements an
<i>[PropertyQuery]</i>		
Enthält vollständige Property SQL-Query		

<i>[PropertyDefinition]</i>			
<b>PropertyID</b> (TAB)	<b>Description</b> (TAB)	<b>DefaultValue</b> (TAB)	<b>Datatype</b>
- <b>PropertyID</b>	Ganzzahl	Property-ID gemäß OPC-DA Spezifikation	
- <b>Description</b>	Text	Beschreibung gemäß OPC-DA Spezifikation	
- <b>DefaultValue</b>	Variante	Standardwert, wenn Wert nicht in Datenbank vorhanden, oder NULL ist. Für NULL leere Angabe.	
- <b>Datatype</b>	Datentyp	Datentyp gemäß OPC-DA Spezifikation. Bei VT_EMPTY wird Datentyp aus Datenbank ermittelt.	
<i>[OveragedDataDetection]</i>			
<b>OveragedDataDetectionEnabled</b>	0/1	Legt fest, ob die Signalausfallerkennung aktiviert (1), oder deaktiviert (0) ist.	
<b>MaxValidDataAge</b>	Ganzzahl [Millisekunden]	Legt das maximal zulässige Datenalter fest.	
<b>DefaultQuality</b>	Ganzzahl/OPC-Quality	Legt fest, welche Signal-Quality bei aktuellen Werten vergeben wird, wenn keine sonstige Übersetzung vorhanden ist. Wert gemäß OPC-DA Spezifikation.	
<b>OveragedDataQuality</b>	Ganzzahl/OPC-Quality	Legt fest, welche Signal-Quality bei alten Werten vergeben wird, wenn keine sonstige Übersetzung vorhanden ist. Wert gemäß OPC-DA Spezifikation.	
<i>[OveragedDataDetection.Conversions]</i>			
<b>SourceQuality</b> (TAB)	<b>FreshDataQuality</b> (TAB)	<b>OveragedDataQuality</b>	
- <b>SourceQuality</b>	Variante	Signal-Qualität aus Datenbank.	
- <b>FreshDataQuality</b>	Ganzzahl/OPC-Quality	Signal-Quality bei aktuellen Werten. Wert gemäß OPC-DA Spezifikation.	
- <b>OveragedDataQuality</b>	Ganzzahl/OPC-Quality	Signal-Quality bei alten Werten. Wert gemäß OPC-DA Spezifikation.	
<i>[ElementFunctions]</i>			
<b>Name</b> (TAB)	<b>ItemType</b> (TAB)	<b>Function</b> (TAB)	<b>Annotation</b>
- <b>Name</b>	Text	Name der Elementfunktion	

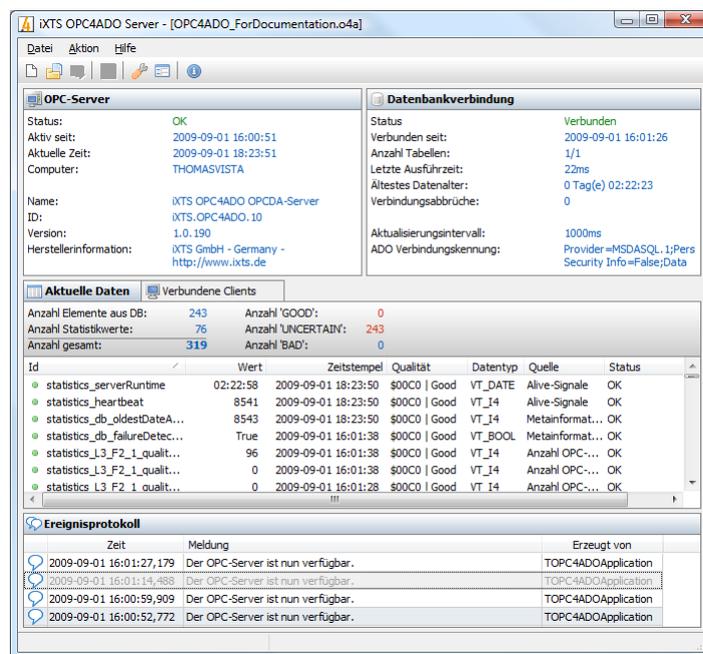
- <b>ItemType</b>	Ganzzahl	Element-Type auf den Elementfunktion angewendet wird.
- <b>Function</b>	Text/intern, Aufzählung	Term der Funktion – dient zur Identifizierung der konkreten Funktion.
- <b>Annotation</b>	Text	Beliebige Anmerkung

# Benutzerschnittstelle

## Das Hauptfenster

Das Hauptfenster von OPC4ADO stellt alle für den Betrieb relevanten Informationen zur Verfügung. Darüber hinaus können alle Einstellungen von diesem Fenster aus erreicht werden.

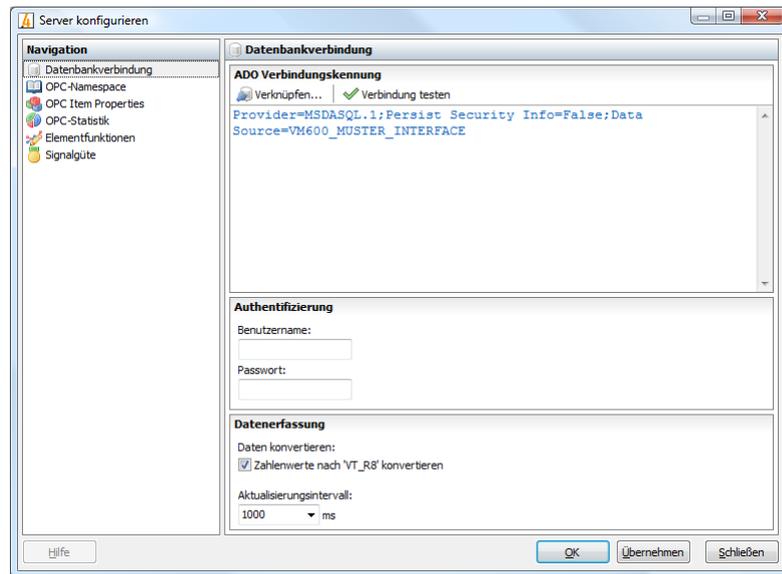
Über den Befehl **Aktion/Server konfigurieren** (🔧) erreichen Sie die Serverkonfiguration. Allgemeine Einstellungen können über den Befehl **Aktion/Globale Programmeinstellungen** (⚙️) geändert werden.



## Serverkonfiguration

Mit Hilfe des Dialogs zur Serverkonfiguration ist es Ihnen möglich, die Anpassungen an Ihre Datenbank durchzuführen. Dieser Dialog ist in mehrere Seiten gegliedert, die Sie über die Navigation links erreichen können.

Änderungen werden erst durch Bestätigung über OK oder Übernehmen durchgeführt.



## Seite Datenbankverbindung

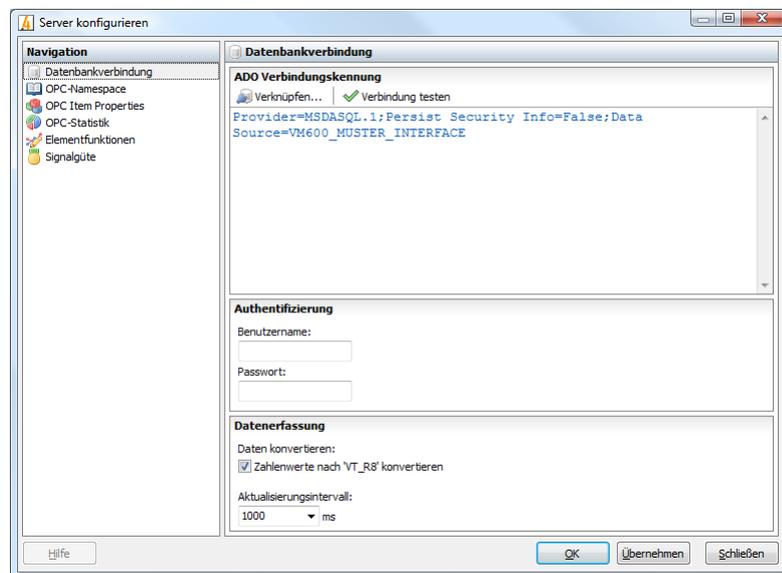
Richten Sie auf der Seite Datenbankverbindung zunächst die Verbindung zum Datenbankserver ein. Klicken Sie hierzu auf **Verknüpfen** (🔗), um das Zusammenbauen der ADO-Verbindungskennung zu vereinfachen.

Geben Sie – wenn erforderlich – Benutzername und Kennwort für die Datenbankverbindung ein.

Durch die Option **Zahlenwerte nach VT\_R8 konvertieren** können Sie festlegen, ob Zahlenwerte aus der Datenbank vereinheitlicht in 64-Bit-Fließkommazahlen (VT\_R8) konvertiert, oder ob die ursprünglichen Datentypen aus der Datenbank beibehalten werden sollen.

Über das **Aktualisierungsintervall** legen Sie fest, in welchem Takt neue Werte aus der Datenbank gelesen werden.

Klicken Sie auf **Verbindung testen** (✅) um zu prüfen, ob die Verbindung zur Datenbank ohne Probleme hergestellt werden kann. Eine entsprechende Erfolgs- oder Fehlermeldung können Sie am oberen Rand des Fensters ablesen.



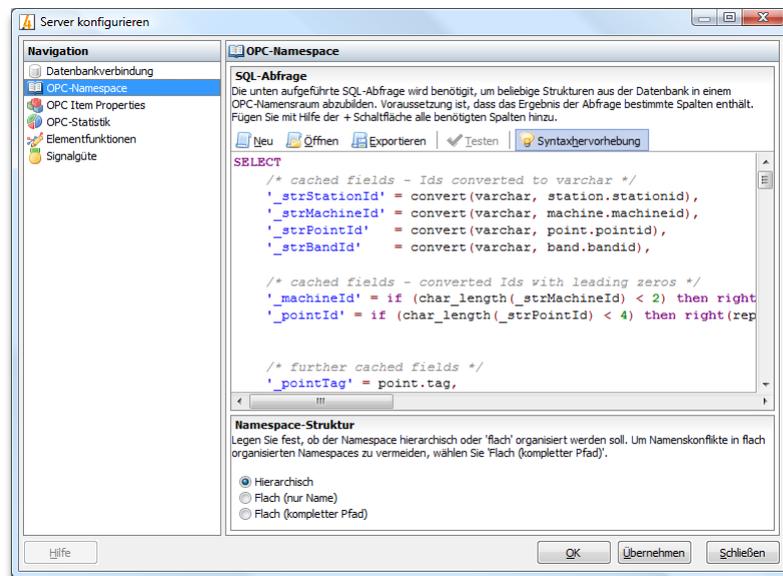
## Seite OPC-Namespace

Auf der Seite OPC-Namespace können Sie die Namespace SQL-Abfrage angeben, sowie die Struktur des Namensraums festlegen.

Für die Definition der SQL-Abfrage stehen Ihnen die gängigen Dateiverwaltungsfunktionen Neu, Öffnen und Exportieren zur Verfügung. Nutzen Sie z.B. **Öffnen** (📄) um eine Textdatei mit der entsprechenden SQL-Abfrage zu laden. Sollte Ihre verwendete Datenbank eine andere SQL-Syntax verwenden, sodass die farblichen Hervorhebungen eher Verwirrung als Ordnung schaffen, können Sie die Syntaxhervorhebung durch Klicken der Schaltfläche **Syntaxhervorhebung** (💡) deaktivieren.

Über die **Namespace-Struktur** legen Sie fest, ob die in der SQL-Abfrage ermittelten Pfadangaben entsprechend übernommen, oder nur die Namen der OPC-Elemente in einem *flach organisierten* Namensraum verwendet werden sollen. Sollten Elemente in verschiedenen Knoten gleiche Namen haben, wählen Sie die Option **Flach (kompletter Pfad)** um Namenskonflikte durch Duplikate zu vermeiden.

Erfahren Sie mehr zum Thema Namespace SQL-Abfrage im Kapitel [Datenbankschnittstelle und -kopplung](#) im Allgemeinen und [Format der Namespace SQL-Abfrage](#) im Konkreten.



## Seite OPC-Item-Properties

Auf der Seite OPC-Item-Properties können Sie zunächst alle verwendeten OPC-Item-Properties definieren und dann mittels Property SQL-Query den OPC-Elementen aus der Namespace SQL-Query (oder den Elementfunktionen) konkrete Daten für die einzelnen OPC-Properties zuweisen.

Fügen Sie neue OPC-Properties hinzu, oder entfernen Sie nicht mehr benötigte mit den Schaltflächen **Hinzufügen** (📄) und **Entfernen** (🗑️).

Jede OPC-Property besitzt eine eindeutige **Property ID**. Beachten Sie hierbei die Regelungen in der OPC-DA Spezifikation. Benutzerdefinierte OPC-Properties sollten eine ID größer 5000 erhalten. **Die OPC-Properties mit den IDs 1 bis 6 werden von OPC4ADO standardmäßig belegt und können nicht überschrieben werden.**

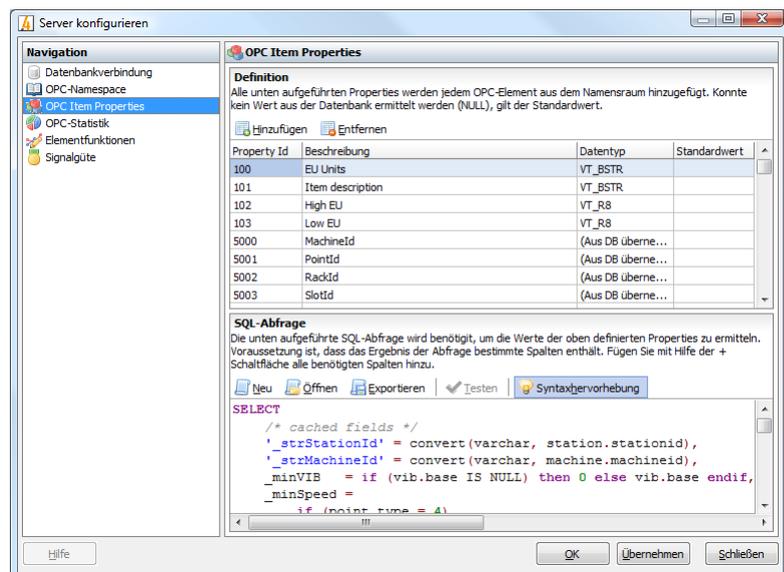
Jede OPC-Property wird durch eine **Beschreibung** näher spezifiziert. Eigens eingeführte Properties sollten eine *sprechende* Beschreibung erhalten. *Bitte beachten Sie die Regelungen der OPC-DA Spezifikation bezüglich der Vorgaben der Beschreibungen der OPC-Properties mit IDs zwischen 100 und 4999.*

OPC-Properties besitzen Werte eines bestimmten **Datentyps**. Geben Sie einen konkreten Datentyp an, oder lassen Sie den Datentyp von OPC4ADO anhand der Meta-Informationen aus der Datenbank automatisch ermitteln. *Bitte beachten Sie die Regelungen der OPC-DA Spezifikation bezüglich der Vorgaben der Datentypen der OPC-Properties mit IDs zwischen 100 und 4999.*

Ist nicht sichergestellt, dass für alle OPC-Elemente auch konkrete Werte für eine OPC-Property aus der Datenbank ermittelt werden können, kann der fehlende Wert mithilfe des **Standardwerts** ersetzt werden. Der Standardwert gilt immer dann, wenn in der Property SQL-Abfrage keine für die Property geltende Spalte vorhanden, oder der Wert der Spalte NULL ist. Ist der Standardwert wiederum NULL (leere Angabe), wird das OPC-Element nicht mit der Property behaftet.

Für die Definition der SQL-Abfrage stehen Ihnen die gängigen Dateiverwaltungsfunktionen Neu, Öffnen und Exportieren zur Verfügung. Nutzen Sie z.B. **Öffnen** (📄) um eine Textdatei mit der entsprechenden SQL-Abfrage zu laden. Sollte Ihre verwendete Datenbank eine andere SQL-Syntax verwenden, sodass die farblichen Hervorhebungen eher Verwirrung als Ordnung schaffen, können Sie die Syntaxhervorhebung durch Klicken der Schaltfläche **Syntaxhervorhebung** (💡) deaktivieren.

Erfahren Sie mehr zum Thema Property SQL-Abfrage im Kapitel [Datenbankschnittstelle und -kopplung](#) im Allgemeinen und [Format der Property SQL-Query](#) im Konkreten.



## Seite OPC-Statistik

Auf der Seite OPC-Statistik können Sie bestimmen, welche Statistikelemente wie im Namensraum erzeugt werden.

Mit der Option **Statistikwerte publizieren** legen Sie fest, ob die [Zählelemente](#) im Namensraum erzeugt werden, oder nicht.

Geben Sie ein **Präfix** an, der sowohl der OPC-ID, also auch dem Namen im OPC-Namespace vorangestellt wird. Durch die Verwendung von Backslashes \ werden Unterknoten erzeugt. So können z.B. alle Statistikelemente durch Angabe von **metastatistics** im Unterknoten „meta“ gesammelt werden, während alle Elemente zusätzlich das Präfix „statistics“ erhalten.

Durch die Option **Anonyme Namen erzeugen** legen Sie fest, ob die Benennung der Zählerelemente an einer Nummerierung (anonym) ausgelegt werden soll, oder die Benennung die übergeordneten Ordner mit einbezieht.

Das Format der anonymen Benennung ist wie folgt: *[Präfix]\_L<Ebene>\_[F[<Index des Knotens>]]<Name des Statistikelements>*.

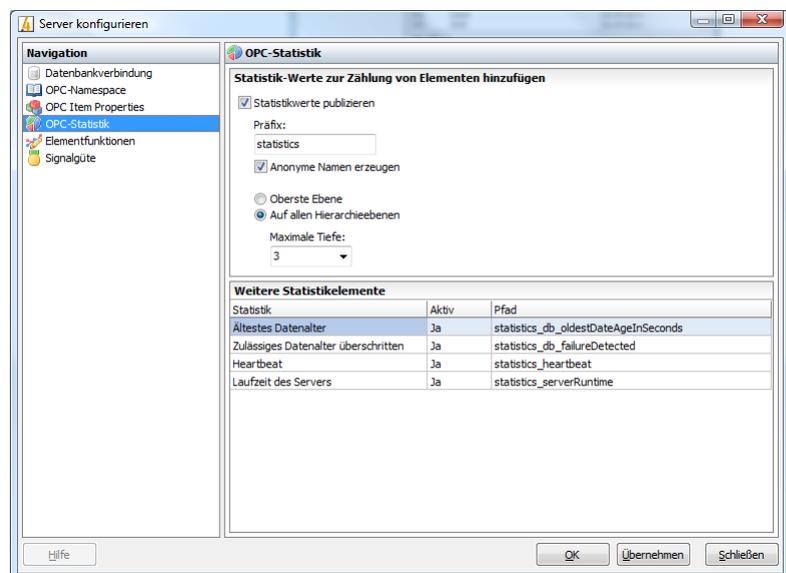
Das Element „Anzahl aller Elemente“ auf oberster Ebene wird z.B. wie folgt benannt: *statistics\_L1\_leafcount*.

Das Element „Anzahl aller Element mit Quality BAD“ im ersten Knoten, dritter Unterknoten wird entsprechend benannt: *statistics\_L3\_F1\_3\_qualitycount\_BAD*.

Durch die Option **oberste Ebene** oder **auf allen Hierarchieebenen** legen Sie fest, ob die Zählerelemente nur auf oberster Ebene im OPC-Namespace, oder in allen Unterknoten erzeugt werden sollen. Durch Angabe der **maximalen Tiefe** können Sie dieses Verhalten zusätzlich beschränken. Die in den Unterknoten erzeugten Elemente *zählen* alle OPC-Elemente ab entsprechenden Knoten.

Unter **weitere Statistikelemente** können Sie zusätzliche Statistikelemente wie Statusinformationen und Lebenszeichen in den OPC-Namespace erzeugen lassen. Setzen Sie **Aktiv** auf **Ja**, damit das entsprechende Statistikelement unter der Pfadangabe hinter **Pfad** im OPC-Namensraum erzeugt wird.

Erfahren Sie mehr über dieses Thema im Kapitel [Statistiken und Lebenszeichen](#).



## Seite Elementfunktionen

Auf der Seite „Elementfunktionen“ können Sie die [Elementfunktionen](#) deklarieren.

Fügen Sie neue Deklarationen hinzu, oder entfernen Sie nicht mehr benötigte mit den Schaltflächen **Hinzufügen** (  ) und **Entfernen** (  ).

Vergeben Sie zunächst einen eindeutigen **Namen**. Der Name wird als Präfix für die zusätzlichen Spalten der Namespace und Property SQL-Query verwendet.

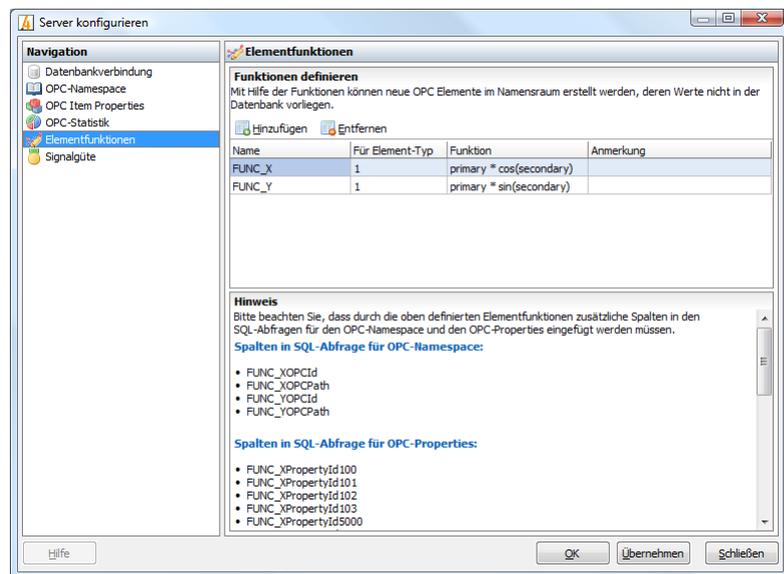
Legen Sie fest, auf welche Typen von [Datensätze der Namespace SQL-Query](#) die Elementfunktion angewendet werden soll. Bestimmen Sie hierfür den **ElementTyp**.

Wählen Sie aus der Liste der verfügbaren **Funktionen** die benötigte Berechnungsvorschrift.

Nutzen Sie das freie Feld unter **Anmerkung**, damit Sie etwaige Kommentare für sich speichern können. Diese Angabe ist beliebig und wird nicht ausgewertet.

Nutzen Sie die automatisch generierten **Spaltennamen** unter **Hinweise**, um im nächsten Schritt die Namespace und Property SQL-Query zu ergänzen.

Erfahren Sie mehr zur Funktionsweise der Elementfunktionen im Kapitel [Elementfunktionen](#).



## Seite Signalgüte

Auf der Seite „Signalgüte“ können Sie die Funktionsweise der [Signalgüte-Optionen](#) konfigurieren.

Durch die Option **Signalausfallerkennung aktivieren** wird die Signalausfallerkennung ein-, bzw. ausgeschaltet. Das **maximal zulässige Datenalter** wird in Millisekunden angegeben.

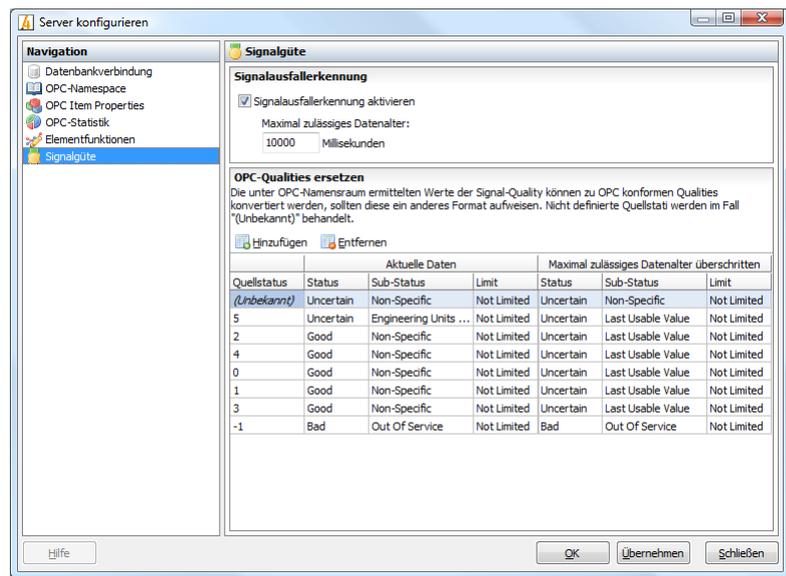
Die Signal-Quality für aktuelle Daten kann in der ersten Zeile (**Unbekannt**) unter **OPC-Qualities ersetzen** angegeben werden. Ist die Signalausfallerkennung aktiviert, kann zusätzlich die Signal-Quality für alte Zeitstempel bestimmt werden.

Fügen Sie neue Quality-Übersetzungen hinzu, oder entfernen Sie nicht mehr benötigte mit den Schaltflächen **Hinzufügen** (  ) und **Entfernen** (  ). Die erste Zeile (Unbekannt) kann nicht gelöscht werden, da diese immer angewendet wird, wenn für einen Status aus der Datenbank keine entsprechende Übersetzung vorhanden ist.

Geben Sie unter **Quellstatus** den Status aus der Datenbank an, für den die Übersetzung durchgeführt werden soll.

Gemäß den Vorgaben der OPC-DA Spezifikation können für eine Signal-Quality neben dem eigentlichen **Status** GOOD, UNCERTAIN und BAD noch **Sub-Status** und **Limits** angegeben werden. Ist die Signalausfallerkennung aktiviert, kann zusätzlich die Signal-Quality für alte Zeitstempel bestimmt werden.

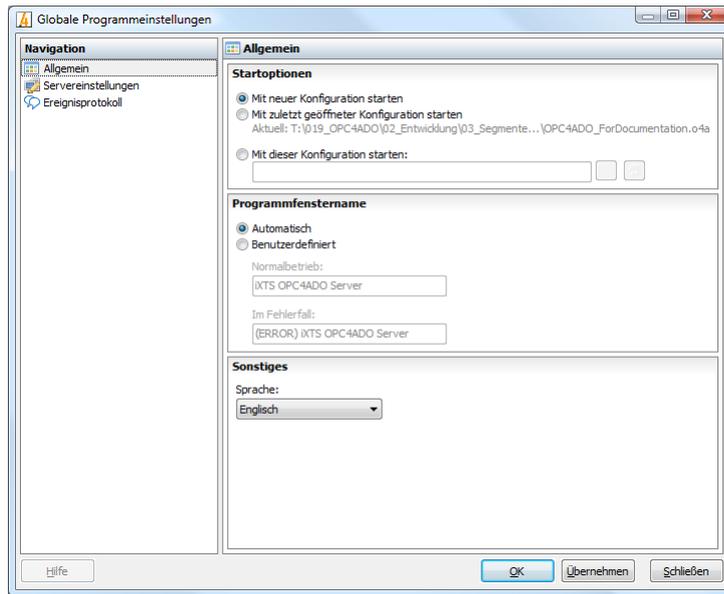
Erfahren Sie mehr zu diesem Thema unter [Umsetzung der Signalgüte](#) im Allgemeinen und [Signalgüte-Optionen](#) im Konkreten.



## Globale Programmeinstellungen

Mit Hilfe des Dialogs zur „globale Programmeinstellungen“ ist es Ihnen möglich, generelle Basiseinstellungen an der Installation vorzunehmen. Dieser Dialog ist in mehrere Seiten gegliedert, die Sie über die Navigation links erreichen können.

Änderungen werden erst durch Bestätigung über OK oder Übernehmen durchgeführt.



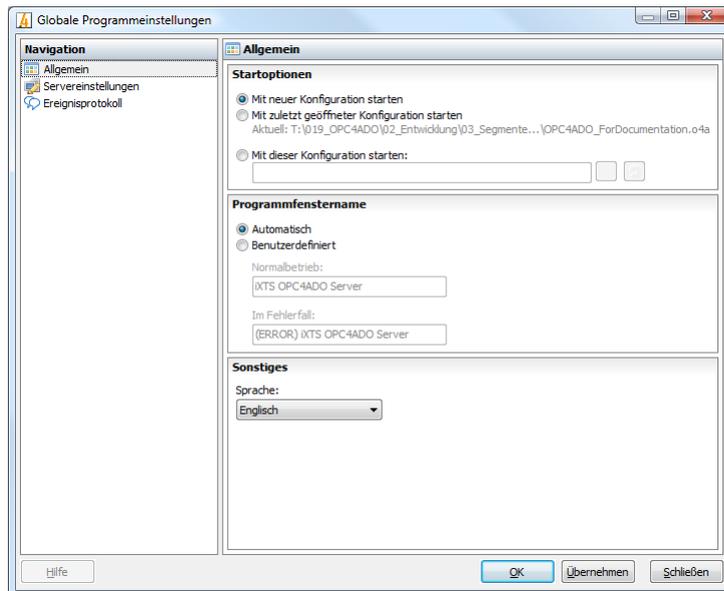
## Seite Allgemein

Auf der Seite „Allgemein“ können grundsätzliche Einstellung getroffen werden.

Mit Hilfe der **Startoptionen** ist es Ihnen möglich, das Verhalten beim Start von OPC4ADO zu beeinflussen.

Möchten Sie bereits am **Programmfenstername** erkennen können, ob in der Konfiguration, oder im laufenden Betrieb ein Fehler vorliegt, können Sie hier den Titel des Hauptfensters entsprechend anpassen.

OPC4ADO liegt in den **Sprachen** Englisch und Deutsch vor.



## Seite Servereinstellungen

Auf der Seite „Servereinstellungen“ können Sie die Angaben zur OPC-DA Serveridentifikation ändern. Bitte beachten Sie, dass diese Änderungen erst nach einem Neustart übernommen werden können.

Bestimmen Sie einen *sprechenden* **OPC Servernamen**. Dieser Name wird in vielen OPC-Clients bei der Serverauswahl angezeigt.

Die **OPC Server-ID** sollte dem Format *Hersteller.Produkt.Version* folgen. Manche OPC-Clients zeigen bei der Serverauswahl statt dem Servernamen die Server-ID an.

Die **OPC Server-GUID** ist eine vom System vergebene, eindeutige Kennung. Dieser *globale unique Identifier* wird vom DCOM-System zur Identifikation dieses Servers verwendet und kann nicht geändert werden.

Die **Herstellerinformationen** können beliebig geändert werden.

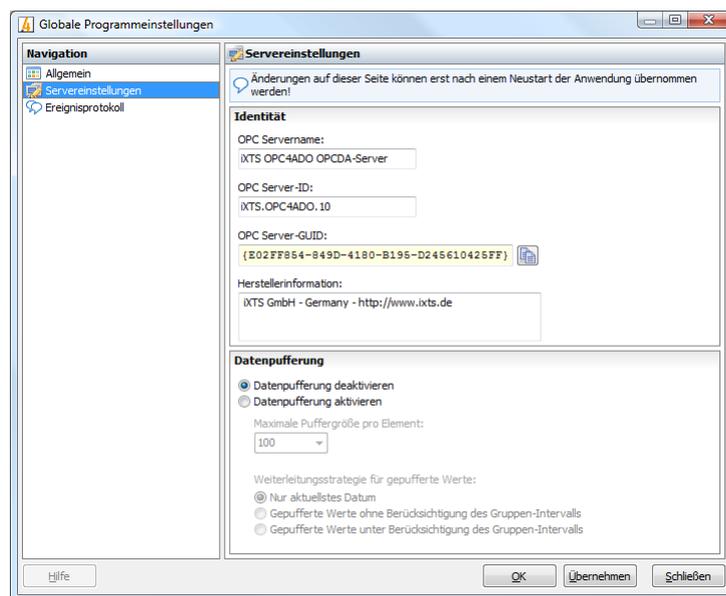
Über die [Datenpufferung](#) können Sie den internen Datencache, der alle über OPC kommunizierten Daten lokal zwischenspeichert, beeinflussen.

Wenn sie die **Datenpufferung aktivieren**, werden für alle OPC-Elemente FIFO-Puffer mit einer maximalen Länge verwaltet. Diese kann über **maximale Puffergröße pro Element** festgelegt werden.

Bei **Weiterleitungsstrategie nur aktuellstes Datum** werden den angeschlossenen Clients nur die aktuellsten Daten übermittelt. Die Option **gepufferte Werte ohne Berücksichtigung des Gruppen-Intervalls** bewirkt, dass alle die seit der letzten Aktualisierung neu gepufferten Daten an die Clients übermittelt werden.

Die Option **gepufferte Werte unter Berücksichtigung des Gruppen-Intervalls** wird derzeit nicht unterstützt.

Erfahren Sie mehr zum Thema Datenpufferung im Kapitel [Datencache und lokale Pufferung](#).



## Seite Ereignisprotokoll

Auf der Seite „Ereignisprotokoll“ können Sie das Verhalten der integrierten Ereignisprotokollierung steuern.

Das Ereignisprotokoll dient dazu, relevante Informationen, d.h. insbesondere Fehlersituationen und Warnung während der Programmlaufzeit aufzuzeichnen. Je nach Situation kann es wünschenswert sein, mehr oder weniger detaillierte Aufzeichnungen zu erhalten. Dies kann durch Veränderung der Einstellung **Detailliertheitsgrad der Ereignisprotokollie-**

**rung** kontrolliert werden. Insgesamt sind sechs verschiedene Einstellungen möglich: *Aus*, *Sehr gering*, *Gering*, *Normal*, *Erweitert* und *Vollständig*. Bei der Voreinstellung *Normal* werden dabei alle wesentlichen Informationen protokolliert, ohne das Protokoll mit irrelevanten Details zu überfluten. Diese Einstellung muss normalerweise nur für Diagnosezwecke geändert werden. Weitere Informationen zu diesem Thema finden sich im Abschnitt [Ereignisprotokoll](#).

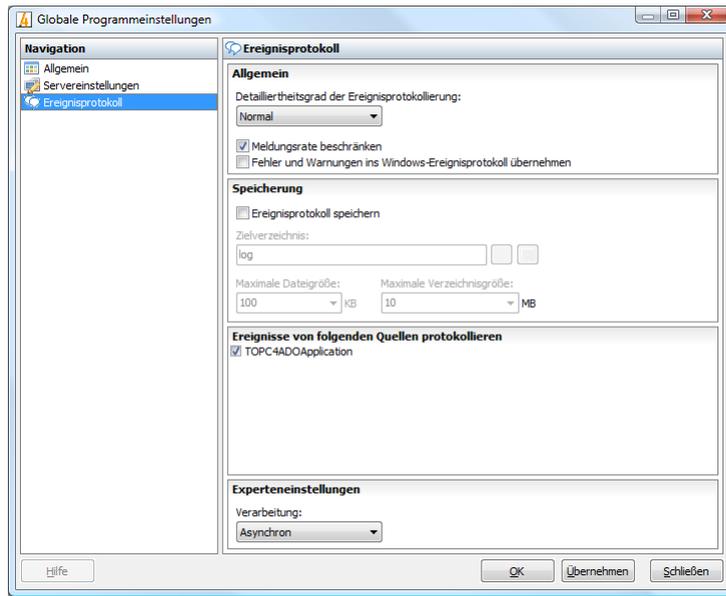
Die Checkbox **Meldungsrate beschränken** ist standardmäßig markiert, um im Falle einer extrem hohen Meldungsrate (z.B. aufgrund einer Fehlfunktion oder eine hohen OPC Last bei gleichzeitig hohem eingestellten Detailliertheitsgrad der Protokollierung) ein „Hängenbleiben“ der Anwendung durch ständig aufeinander folgendes Neuzeichnen zu verhindern. Dazu werden gegebenenfalls Meldungen verworfen. Diese Option sollte nur dann abgeschaltet werden, wenn das Verwerfen von Ereignissen auf jeden Fall verhindert werden soll. Dies kann allerdings unter Umständen zu dem oben beschriebenen Verhalten führen.

Die Bedeutung der Checkbox **Fehler und Warnungen ins Windows-Ereignisprotokoll übernehmen** dürfte selbsterklärend sein. Diese Option ist standardmäßig aktiviert.

Auf Wunsch können die protokollierten Ereignisse auch automatisch auf Festplatte abgelegt werden, indem die Option **Ereignisprotokoll auf Datenträger speichern** aktiviert wird. In diesem Fall muss ein Verzeichnis für die Protokolldateien, sowie eine maximale Datei- und Verzeichnisgröße angegeben werden. Bei Erreichen der maximalen Dateigröße wird die aktuelle Datei abgeschlossen und eine neue Datei angelegt. Beim Erreichen der maximalen Verzeichnisgröße wird die älteste Datei automatisch gelöscht.

Eine andere Möglichkeit die Anzahl der aufgezeichneten Ereignisse zu limitieren (insbesondere bei einem hohen Detailliertheitsgrad), besteht darin die Einstellungen für die Option **Ereignisse von folgenden Quellen protokollieren** anzupassen. Durch Entfernen der Markierung für einzelne Quellen werden diese von der Protokollierung ausgeschlossen. Die Liste der Quellen wird dynamisch erstellt, d.h. beim ersten Programmstart ist die Liste leer und erst mit dem Eintreffen der Nachrichten, werden die zugehörigen Ereignisquellen hinzugefügt. Standardmäßig sind alle Quellen berechtigt, Ereignisse zu generieren und ein Ändern dieser Einstellungen sollte nur zu Diagnosezwecken notwendig sein. Weitere Informationen entnehmen Sie bitte dem Abschnitt [Ereignisprotokoll](#).

Die **Experteneinstellungen** sollten im Normalfall nicht geändert werden.



# Diagnose

---

## Ereignisprotokoll

Die Applikation generiert eine Reihe von so genannten Ereignissen, die Informationen über den aktuellen Systemzustand, aufgetretene Fehler oder andere Details enthalten. Diese Ereignisse werden innerhalb der Applikation immer im Ereignisprotokoll aufgezeichnet und können zusätzlich bei Bedarf automatisch auf einem Datenträger abgelegt werden. Wenn ein Problem innerhalb des Betriebs auftritt, können die Protokoll-dateien von erheblichem Nutzen für die Fehlersuche sein. Standardmäßig ist das automatische Speichern der Ereignisse nicht aktiviert und muss daher bei Bedarf erst aktiviert werden. Ebenso wird standardmäßig nur eine begrenzte Auswahl von Informationen protokolliert, da der Detailliertheitsgrad auf Normal gesetzt ist. Beide Einstellungen können auf Einstellungsseite des Ereignisprotokolls geändert werden.

Je höher der Detailliertheitsgrad ist, desto mehr Informationen werden aufgezeichnet. Es ist jedoch zu beachten, dass die Leistungsfähigkeit bei hohem Detailliertheitsgrad nachteilig beeinflusst werden kann. Die empfohlene Einstellung ist **Normal**. In den meisten Fällen sollte eine Erhöhung auf **Erweitert** ausreichend sein, wenn nicht, so können durch Verwendung der Einstellung **Vollständig** alle verfügbaren Informationen aufgezeichnet werden.

Bei Verwendung der Einstellungen **Sehr gering** und **Gering** zeichnet die Applikation alle Informationen und Fehler auf, die für den Benutzer von unmittelbarem Interesse sind. Bei Verwendung der **Normal**-Einstellungen werden zusätzlich nützliche und hilfreiche Informationen mit aufgezeichnet. Zusätzliche Informationen von untergeordneter Bedeutung werden bei der Einstellung **Erweitert** aufgezeichnet und die Einstellung **Vollständig** dient insbesondere dem geschulten Support-Mitarbeiter dazu, weitere Rückschlüsse über interne Programmzustände zu ziehen.

Insbesondere bei einem hohen Detailliertheitsgrad der Ereignisprotokollierung kann es zu einer sehr hohen Ereignisrate und damit zu einer sehr hohen Serverlast kommen. Um diesen Effekt zu vermeiden ist es möglich, einzelne Softwareklassen von der Generierung von Ereignissen auszuschließen. Um diese Option sinnvoll einsetzen zu können ist ein tieferes Verständnis über die internen Programmvorgänge nötig, so dass unter normalen Umständen alle Klassen als Ereignisquellen aktiviert bleiben sollten.

# Index

## A

Aktivierung 6  
Aufbau dieses Handbuchs 2

## B

Benutzerschnittstelle 24

## D

Das Hauptfenster 24  
Dateien 17  
Datenbankschnittstelle und -kopplung 10  
Datencache und lokale Pufferung 9  
Diagnose 35  
Die Technik dahinter 8

## E

Einleitung 2  
Elementfunktionen 16  
Ereignisprotokoll 35

## F

Format der Namespace SQL-Query 11  
Format der Property SQL-Query 13

## G

Globale Programmeinstellungen 30

## H

Hardwarevoraussetzungen 4

## I

Implizite Zuweisung der Signal-Qualities 14  
Inbetriebnahme 4  
Installation 4

## K

Konfigurationsdatei (.o4a) 20

## L

Lizenzierung 5  
Lizenzierung und Aktivierung 4

## O

OPC Standardkonformität 3

## P

Parallele Installation mehrerer Server 10  
Programmoptionen (.ini) 18

## R

Registrierung und Deregistrierung einer  
Installation 9

## S

Seite Allgemein 31  
Seite Datenbankverbindung 25  
Seite Elementfunktionen 29  
Seite Ereignisprotokoll 32  
Seite OPC-Item-Properties 26  
Seite OPC-Namespace 26  
Seite OPC-Statistik 27  
Seite Servereinstellungen 31  
Seite Signalgüte 29  
Serverkonfiguration 24  
Signalgüte-Optionen 15  
Softwarevoraussetzungen 4  
Statistiken und Lebenszeichen 15  
Systembeschreibung 7  
Systemvoraussetzungen 4

## T

Trial Version 3

## U

Überblick 7  
Umsetzung der Signalgüte (Signal-Quality) 14

## V

Verfügbare Funktionen 16  
Vorgehen beim Deklarieren einer  
Elementfunktion 17

## W

Wann wird OPC4ADO benötigt 7  
Was ist der OPC Server for ADO (OPC4ADO) 2  
Weitere Quellen 3  
Wer sollte dieses Handbuch lesen 2

