

WinAC XML Data Access

Anwenderdokumentation V 1.0

WinAC • August 2013

Applikationen & Tools

Answers for industry.

SIEMENS

Siemens Industry Online Support

Dieser Beitrag stammt aus dem Siemens Industry Online Support. Durch den folgenden Link gelangen Sie direkt zur Downloadseite dieses Dokuments.

<http://support.automation.siemens.com/WW/view/de/78028682>

Bei Fragen zu diesem Beitrag wenden Sie sich bitte über folgende E-Mail-Adresse an uns:

online-support.automation@siemens.com

applications.aud.koe.nrh.rd@siemens.com

SIEMENS

SIMATIC WinAC XML Data Access

Anwenderdokumentation

Installation

1

Funktionelle
Beschreibung

2

Funktionsbausteine

3

Fehlermeldungen

4

Literaturhinweis

5

Historie

6

Gewährleistung und Haftung

Hinweis

Die Applikationsbeispiele sind unverbindlich und erheben keinen Anspruch auf Vollständigkeit hinsichtlich Konfiguration und Ausstattung sowie jeglicher Eventualitäten. Die Applikationsbeispiele stellen keine kundenspezifischen Lösungen dar, sondern sollen lediglich Hilfestellung bieten bei typischen Aufgabenstellungen. Sie sind für den sachgemäßen Betrieb der beschriebenen Produkte selbst verantwortlich. Diese Applikationsbeispiele entheben Sie nicht der Verpflichtung zu sicherem Umgang bei Anwendung, Installation, Betrieb und Wartung. Durch Nutzung dieser Applikationsbeispiele erkennen Sie an, dass wir über die beschriebene Haftungsregelung hinaus nicht für etwaige Schäden haftbar gemacht werden können. Wir behalten uns das Recht vor, Änderungen an diesen Applikationsbeispielen jederzeit ohne Ankündigung durchzuführen. Bei Abweichungen zwischen den Vorschlägen in diesem Applikationsbeispiel und anderen Siemens Publikationen, wie z.B. Katalogen, hat der Inhalt der anderen Dokumentation Vorrang.

Für die in diesem Dokument enthaltenen Informationen übernehmen wir keine Gewähr.

Unsere Haftung, gleich aus welchem Rechtsgrund, für durch die Verwendung der in diesem Applikationsbeispiel beschriebenen Beispiele, Hinweise, Programme, Projektierungs- und Leistungsdaten usw. verursachte Schäden ist ausgeschlossen, soweit nicht z.B. nach dem Produkthaftungsgesetz in Fällen des Vorsatzes, der groben Fahrlässigkeit, wegen der Verletzung des Lebens, des Körpers oder der Gesundheit, wegen einer Übernahme der Garantie für die Beschaffenheit einer Sache, wegen des arglistigen Verschweigens eines Mangels oder wegen Verletzung wesentlicher Vertragspflichten zwingend gehaftet wird. Der Schadensersatz wegen Verletzung wesentlicher Vertragspflichten ist jedoch auf den vertragstypischen, vorhersehbaren Schaden begrenzt, soweit nicht Vorsatz oder grobe Fahrlässigkeit vorliegt oder wegen der Verletzung des Lebens, des Körpers oder der Gesundheit zwingend gehaftet wird. Eine Änderung der Beweislast zu Ihrem Nachteil ist hiermit nicht verbunden.

Weitergabe oder Vervielfältigung dieser Applikationsbeispiele oder Auszüge daraus sind nicht gestattet, soweit nicht ausdrücklich von Siemens Industry Sector zugestanden.

Inhaltsverzeichnis

Gewährleistung und Haftung	4
1 Automatisierungsaufgabe	6
1.1 Übersicht	6
1.2 Erforderliche Kenntnisse	6
1.3 Verwendete Hard- und Software-Komponenten	7
2 Installation	8
2.1 Quickstart	8
2.2 Installation des WinAC-Treibers auf dem Runtime-Rechner	9
2.3 Installation des WinAC-Treibers auf dem Entwicklungsrechner mit STEP 7 Classic	9
2.4 Installation des WinAC-Treibers auf dem Entwicklungsrechner mit STEP 7 V11/V12 (TIA-Portal)	10
2.5 Versionsstand des Treibers.....	11
3 Funktionelle Beschreibung	12
3.1 Gesamtübersicht	12
3.2 Funktionsumfang.....	13
4 Funktionsbausteine	14
4.1 Konzept der STEP 7 FBs	14
4.2 FBXMLInit.....	14
4.3 FBXMLDocLoadFile	15
4.3.1 Zusammenhang REQ, BUSY, DONE und ERROR	15
4.4 FBXMLDocSaveFile	18
4.5 FBXMLGet.....	19
4.5.1 Datentypen	20
4.6 FBXMLSet	21
5 Fehlermeldungen	23
5.1 Fehlermeldungen vom WinAC ODK	23
5.2 Spezielle Fehlermeldungen des WinAC XMLDataAccess.....	25
6 Literaturhinweis	26
6.1 Literaturangaben	26
6.2 Internet-Link-Angaben.....	26
7 Historie	26

1 Automatisierungsaufgabe

1.1 Übersicht

Einführung/Einleitung

Es ist mit der WinAC nicht möglich, Daten in XML Dateien zu schreiben bzw. aus XML Dateien zu lesen.

Diese Funktionalität bietet die Erweiterung WinAC XMLDataAccess

1.2 Erforderliche Kenntnisse

Für das Verständnis der vorliegenden Dokumentation werden Kenntnisse vorausgesetzt:

- SIMATIC WinAC RTX 2010
- SIMATIC Manager STEP7 V5.5
oder
- STEP 7 V11/V12(TIA-Portal)

1.3 Verwendete Hard- und Software-Komponenten

Die Applikation wurde mit den nachfolgenden Komponenten erstellt:

Hardware-Komponenten

- Simatic Microbox IPC 427C (1,2 GHz, 4 GB RAM, 4 GB CF-Card) mit Windows XP embedded SP3

Standard Software-Komponenten

- SIMATIC WinAC RTX 2010
- SIMATIC Manager STEP 7 V5.5
- TIA Portal V11
- TIA Portal V12

Beispieldateien und Projekte

Die folgende Liste enthält alle Dateien und Projekte, die in diesem Treiber-Paket verwendet werden.

Komponente	Hinweis
Documentation \ WinAC-XMLDataAccess_v10_DE.pdf Documentation \ WinAC-XMLDataAccess_v10_EN.pdf	Die Dokumentation in Deutsch und Englisch
Driver \ WinAcXMLServer.dll	Der WinAC Treiber
Driver \ setup.bat	Setup für den Treiber
Driver \ test.xml	Test XML Datei
STEP 7_V5_Demo	STEP 7 Classic (V5.5) Beispielprojekt mit allen notwendigen WinAC Funktionsbausteinen
TIA_V11_Demo	TIA Portal V11 Beispielprojekt mit allen notwendigen WinAC Funktionsbausteinen
TIA_V12_Demo	TIA Portal V12 Beispielprojekt mit allen notwendigen WinAC Funktionsbausteinen

Tabelle 1-1 Enthaltende Dateien

2 Installation

2.1 Quickstart

Runtime-System

- Treiber-DLL mit **setup.bat** installieren

Engineering-System STEP 7 V5.5

- STEP 7-Demo-Projekt aus Lieferumfang dearchivieren und öffnen
- Kopieren der Treiber FBs sowie der SFBs in das eigene Projekt
- Funktionsbaustein FBXMLInit integrieren
- Funktionsbaustein FBXMLDocLoadFile mit passendem Dateinamen aufrufen (positive Flanke öffnet)
- Funktionsbausteine FBXMLGet / FBXMLPut für Lesen / Schreiben aufrufen (positive Flanke liest / schreibt Datei)

Engineering-System TIA Portal V11/V12

- STEP 7-Demo-Projekt aus Lieferumfang dearchivieren und öffnen
- Kopieren der Treiber FBs in das eigene Projekt
Projekt muss WinAC enthalten, ansonsten werden Treiber FBs als fehlerhaft markiert.
- Funktionsbaustein FBXMLInit integrieren
- Funktionsbaustein FBXMLDocLoadFile mit passendem Dateinamen aufrufen (positive Flanke öffnet)
- Funktionsbausteine FBXMLGet / FBXMLPut für Lesen / Schreiben aufrufen (positive Flanke liest / schreibt Datei)

2.2 Installation des WinAC-Treibers auf dem Runtime-Rechner

Die Installation des WinAC XMLDataAccess beschränkt sich auf das Kopieren der Treiber -DLL in das System32-Verzeichnis. Dazu gibt es eine BATCH-Datei **setup.bat**.

Installation unter Windows XP (embedded)

Unter Windows XP (embedded) kann man die **setup.bat** einfach ausführen. Das funktioniert auch direkt von einem USB-Stick.

Installation unter Windows 7

Für das Kopieren in das system32-Verzeichnis werden Administrator-Rechte benötigt. Dazu muss man die **setup.bat** als Administrator ausführen (Rechte Maustaste, Ausführen als Administrator). Die Windows 7 UAC muss mit „Ja“ bestätigt werden.

Das funktioniert auch direkt vom USB-Stick.

2.3 Installation des WinAC-Treibers auf dem Entwicklungsrechner mit STEP 7 V5

Auf dem Entwicklungsrechner werden folgende Komponenten benötigt:

- diese Dokumentation
- das STEP 7 (V5.5) - Beispielprojekt

Dem S7-Demo-Projekt können die erforderlichen FBs für das STEP 7 Anwenderprogramm entnommen werden.

Eine Installation des Treibers (**setup.bat**) auf dem Entwicklungsrechner ist NICHT erforderlich.

2.4 Installation des WinAC-Treibers auf dem Entwicklungsrechner mit STEP 7 V11/V12 (TIA-Portal)

Auf dem Entwicklungsrechner werden folgende Komponenten benötigt:

- diese Dokumentation
- das STEP 7 V11/V12 (TIA-Portal) Beispielprojekt

Dem S7-Demo-Projekt können die erforderlichen FBs für das STEP 7 Anwenderprogramm entnommen werden.

Eine Installation des Treibers (**setup.bat**) auf dem Entwicklungsrechner ist NICHT erforderlich.

WinAC - CPU im Projekt

Werden die Treiber-Bausteine in ein Projekt kopiert, welches keine WinAC als CPU enthält, werden die FBs als fehlerhaft markiert, da die benötigten ODK SFBs nicht unterstützt werden.

Kopieren der Komponenten (Bausteine, Anwenderkonstanten und Tabellen)

Für die Verwendung des Treibers im TIA Portal müssen die folgenden Komponenten aus dem Beispielprojekt kopiert werden:

- Treiber-Bausteine (Ordner „ProgrammBausteine“),
- Anwenderkonstanten (Ordner „PLC-Variablen“ → WinAC_XMLDataAccess_Constants),

Zum Kopieren zwischen zwei Projekten wird eine zweite Instanz des TIA Portals benötigt. In dieser zweiten Instanz, wird das Zielprojekt geöffnet und von der ersten Instanz (hier befindet sich das Demo Projekt des XMLDataAccess) per „*drag & drop*“ oder *kopieren (Quellprojekt), Einfügen* in das Zielprojekt kopiert.

Übersetzen der Bausteine

Es kann vorkommen, dass die Funktionen des WinAC ODK (EXEC_COM und CREA_COM) nicht erkannt werden und rot in den Schnittstellen hinterlegt sind. Ist das der Fall, müssen diese Bausteine einmal übersetzt werden, damit diese Funktionen erkannt werden.

2.5 Versionsstand des Treibers

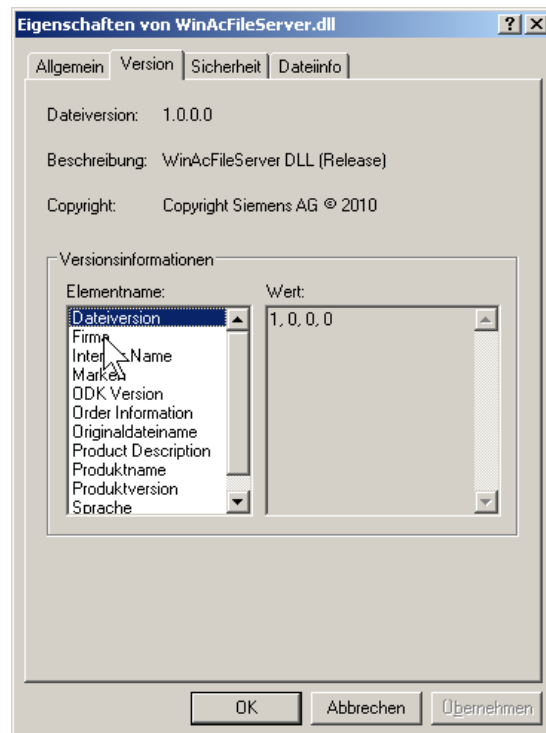
Treiber-Version unter Windows ermitteln

Die registrierte Treiber-DLL liegt im Systemverzeichnis, z.B.

C:\Windows\system32\WinAcXMLServer.dll

Die Version der Treiber DLL kann man im Windows-Explorer über die Dateieigenschaften erfahren (rechte Maustaste → Eigenschaften).

Abbildung 2-1 Versionsstand der Treiber- DLL



3 Funktionelle Beschreibung

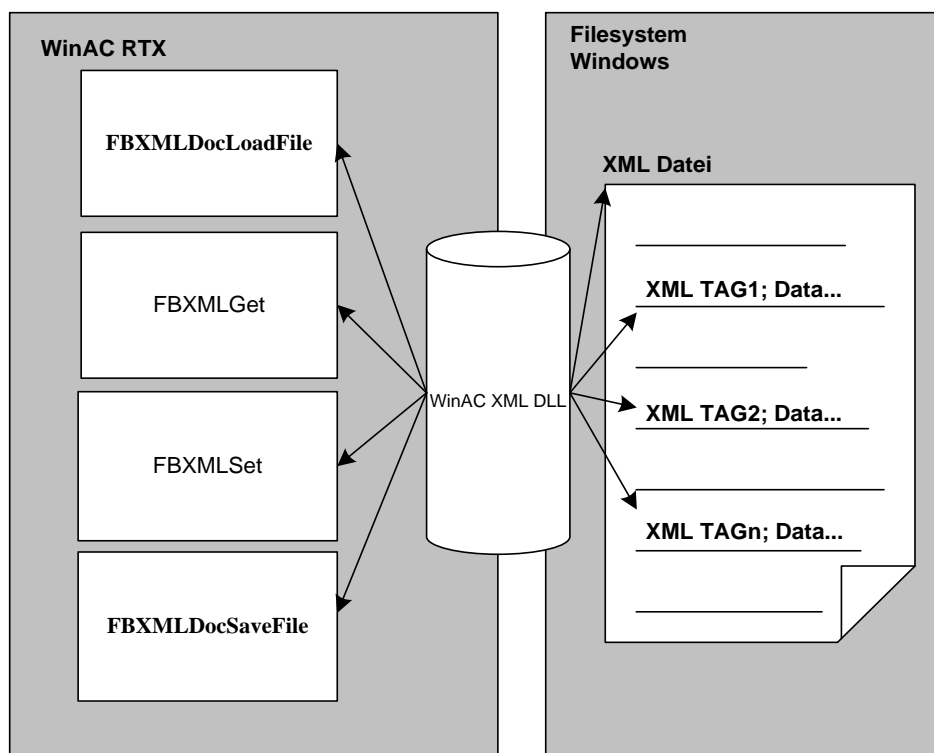
3.1 Gesamtübersicht

Die Lösung besteht aus zwei Teilen:

- STEP 7-FBs zum Lesen und Schreiben von XML Tags
- Einer Treiber-DLL, die diese Funktion realisiert

Der Anwender muss auf seinem Zielsystem (Runtime-System) die DLL installieren. Auf der Engineering-Station (Simatic Manager) werden nur die neuen STEP 7-FBs für das Kunden-Projekt benötigt.

Abbildung 3-1 Struktur der Lösung



3.2 Funktionsumfang

Dieser Treiber wurde für die Anwendungsfälle geschrieben, in denen man beliebige (einzelne) Informationen aus einer XML-Datei lesen oder beliebige (einzelne) Informationen in eine XML-Datei schreiben möchte.

Es geht also nicht darum, die komplette XML-Datei in einen DB zu schreiben, sondern gezielt einzelne Informationen aus der XML Datei zu bearbeiten (lesen/schreiben).

Der Anwender bekommt z.B. vom Anlagenbetreiber eine sehr große XML-Datei mit allen möglichen Informationen. Das WinAC Programm benötigt daraus aber nur ganz spezifische Werte.

In der XML-Welt bietet sich die **XPath** Syntax an, um auf Werte in einer XML-Datei zuzugreifen.

(siehe z.B. http://www.w3schools.com/xpath/xpath_syntax.asp)

Die Daten werden über XPath definiert.

Für das Parsen des XML Dokuments wird TyniXPath benutzt.

Dieses unterliegt der zlib Lizenz:

„Diese Software wird ohne ausdrückliche oder implizierte Garantie bereitgestellt. In keinem Fall können die Autoren für irgendwelche Schäden, die durch die Benutzung dieser Software entstanden sind, haftbar gemacht werden.

Es ist jedem gestattet, diese Software für jeden Zweck, inklusive kommerzieller Anwendungen, zu benutzen, zu verändern und sie frei weiterzuverbreiten, sofern folgende Bedingungen erfüllt sind:

Die Herkunft dieser Software darf nicht falsch dargestellt werden; Sie dürfen nicht angeben, dass Sie die ursprüngliche Software geschrieben haben. Wenn Sie diese Software in einem Produkt benutzen, würde eine Erwähnung geschätzt werden, sie ist aber nicht erforderlich.

Veränderte Quelltextversionen müssen deutlich als solche gekennzeichnet werden und dürfen nicht als die Originalsoftware dargestellt werden.

Diese Notiz darf in den Quelltexten nicht verändert oder gelöscht werden.

4 Funktionsbausteine

4.1 Konzept der STEP 7 FBs

Folgende Funktionen werden realisiert:

- FBXMLInit - Treiber Initialisierung
- FBXMLDocLoadFile - Datei öffnen / parsen
- FBXMLGet - Wert über XPath abfragen
- FBXMLPut - Wert über XPath schreiben
- FBXMLDocSaveFile - Speicher des XML Datei

4.2 FBXMLInit

Mit diesem FB wird die Funktionalität des Treibers initialisiert. Er muss einmalig vor allen anderen FBs des XMLDataAccess aufgerufen werden.

Tabelle 4-1 Parameter des FBs FBXMLInit

Parameter	In/ Out	Typ	Beschreibung
ERROR	Out	Bool	Fehler aufgetreten
STATUS	Out	WORD	Fehler Information
ODK_REF	Out	WORD	Referenz zum Treiber ^{*1)}

Hinweis

^{*1)} Der Wert von **ODK_REF** muss allen anderen FBs des WinAC FileServers als Eingangsparameter mitgegeben werden.

4.3 FBXMLDocLoadFile

Mit diesem FB wird eine XML-Datei geöffnet.

Der Aufruf ist asynchron ausgelegt, d.h. eine Flanke am REQ signalisiert den Auftrag und ein gesetztes „DONE“ signalisiert das Ende der Bearbeitung.

Über ID wird die jeweilige Datei angesprochen. Es werden bis zu 32 Dateien gleichzeitig unterstützt.

Tabelle 4-2 Parameter des FBXMLDocLoadFile

Parameter	In/Out	Typ	Beschreibung
ODK_REF	In	WORD	Referenz zum Treiber (siehe FBFileInit)
ID	In	INT	Ident für diese Datei 0..31
XML_FILE_NAME	In	STRING	Name der Datei (einschl. Pfad)
RESET	In	BOOL	Reset interner Variablen (muss vor erster Benutzung einmalig gesetzt sein)
REQ	In	BOOL	Positive Flanke öffnet die Datei
BUSY	Out	BOOL	Auftrag läuft
DONE	Out	BOOL	Auftrag fertig ohne Fehler
ERROR	Out	BOOL	Fehler aufgetreten
XML_ERROR_CODE	Out	DWORD	XML Fehler Information
ODK_ERROR_CODE	Out	DWORD	ODK Fehler Information

4.3.1 Zusammenhang REQ, BUSY, DONE und ERROR

Eine positive Flanke von REQ startet die Befehlsausführung. Während der Bearbeitung des Befehls steht BUSY auf TRUE. Nach dem Ende der Bearbeitung steht entweder DONE oder ERROR für mindestens einen Aufruf lang an. Wird REQ auf 0 gesetzt, werden auch DONE bzw. ERROR beim nächsten Aufruf zurückgesetzt.

Die folgenden Zeitabläufe verdeutlichen den Zusammenhang:

Abbildung 4-1 Zeitverlauf: REQ steht lange an, Bearbeitung ohne Fehler

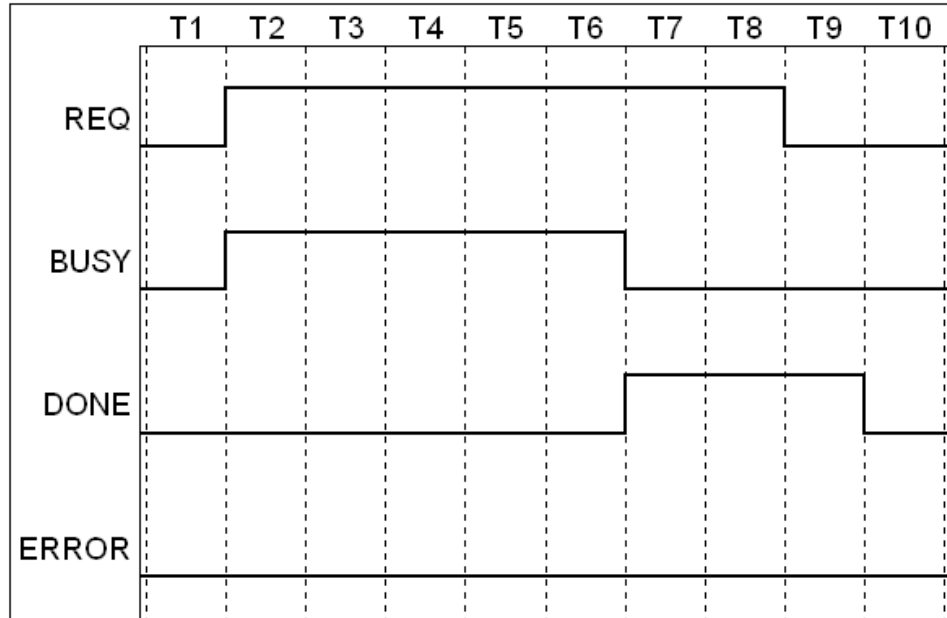


Abbildung 4-2 Zeitverlauf: REQ steht lange an, Bearbeitung mit Fehler

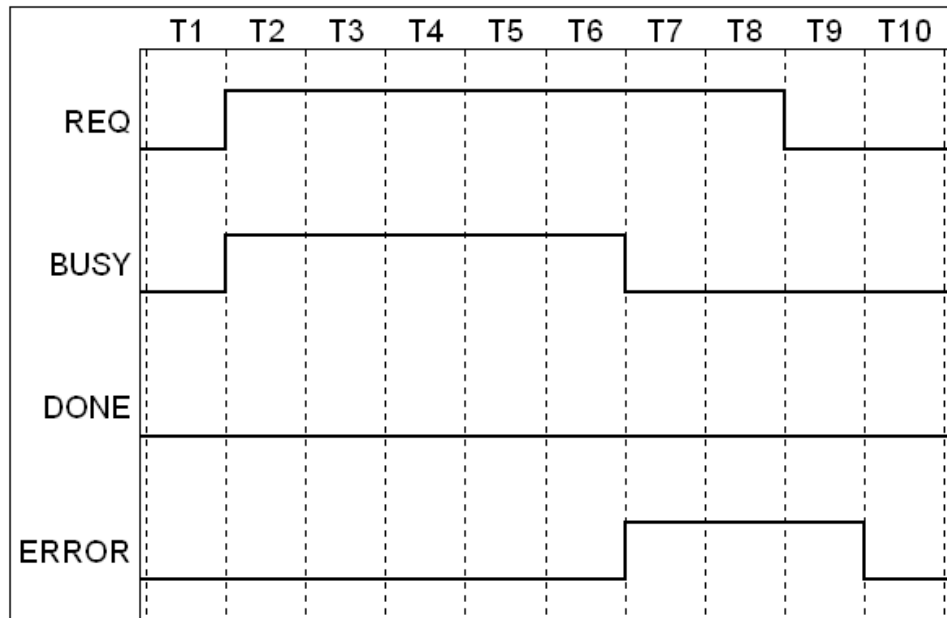


Abbildung 4-3 Zeitverlauf: REQ steht kurz an, Bearbeitung ohne Fehler

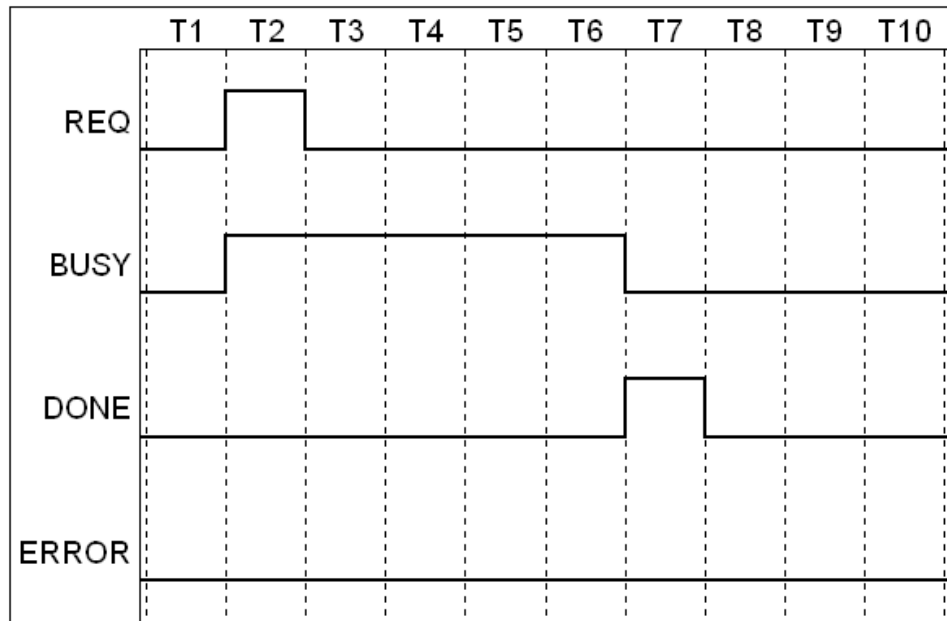
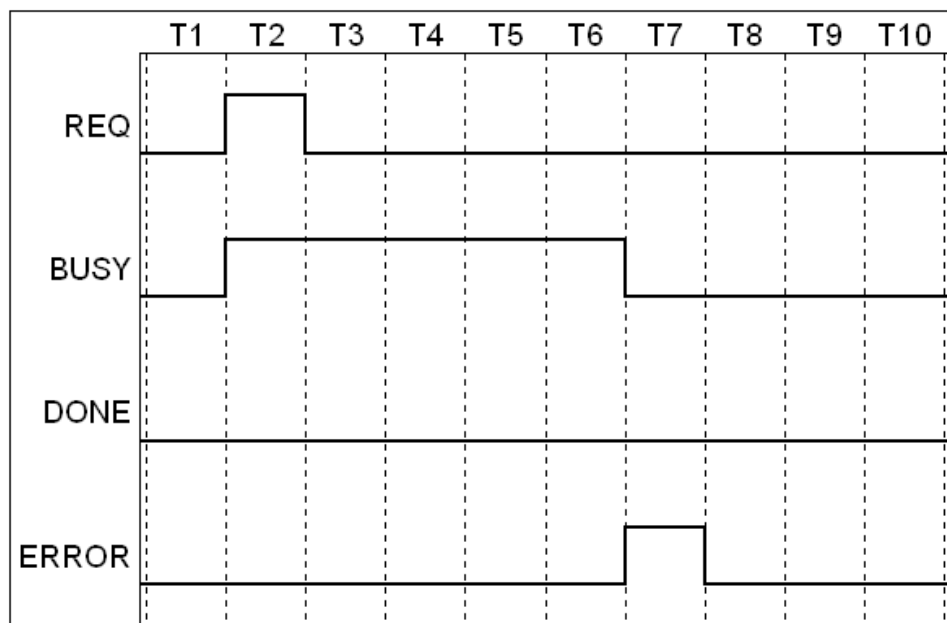


Abbildung 4-4 Zeitverlauf: REQ steht kurz an, Bearbeitung mit Fehler



4.4 FBXMLDocSaveFile

Mit diesem FB wird eine geöffnete XML-Datei gespeichert.

Der Aufruf ist asynchron ausgelegt, d.h. eine Flanke am REQ signalisiert den Auftrag und ein gesetztes „DONE“ signalisiert das Ende der Bearbeitung.

Über ID wird die jeweilige XML-Datei angesprochen. Es werden bis zu 32 Dateien gleichzeitig unterstützt.

Tabelle 4-3 Parameter des FBXMLDocSaveFile

Parameter	In/Out	Typ	Beschreibung
ODK_REF	In	WORD	Referenz zum Treiber (siehe FBFileInit)
ID	In	INT	Ident für diese Datei 0..31
RESET	In	BOOL	Reset interner Variablen (muss vor erster Benutzung einmalig gesetzt sein)
REQ	In	BOOL	Positive Flanke öffnet die Datei
BUSY	Out	BOOL	Auftrag läuft
DONE	Out	BOOL	Auftrag fertig ohne Fehler
ERROR	Out	BOOL	Fehler aufgetreten
XML_ERROR_CODE	Out	DWORD	XML Fehler Information
ODK_ERROR_CODE	Out	DWORD	ODK Fehler Information

Der zeitliche Verlauf von REQ, BUSY, DONE und ERROR ist im Kapitel 4.3.1.

4.5 FBXMLGet

Mit diesem FB werden die XML Daten gelesen und in einer SPS gespeichert.

Die Selektion, welche XML Daten gelesen werden sollen, erfolgt über ein XPath-Ausdruck. Die Selektion mehrerer gleichartiger Elemente ist möglich. Ausgangsseitig wird in diesem Fall ein Array zurückgegeben.

Tabelle 4-4 Parameter des FBXMLGet

Parameter	In/Output	Typ	Beschreibung
ODK_REF	In	WORD	Referenz zum Treiber (siehe FBFileInit)
ID	In	INT	Ident für diese Datei 0..31
XML_ITEM	in	XML_ITEM_TYP	Beschreibung der XML_ITEMS siehe Tabelle 4-3
RESET	In	BOOL	Reset interner Variablen (muss vor erster Benutzung einmalig gesetzt sein)
REQ	In	BOOL	Positive Flanke öffnet die Datei
BUSY	Out	BOOL	Auftrag läuft
DONE	Out	BOOL	Auftrag fertig ohne Fehler
ERROR	Out	BOOL	Fehler aufgetreten
XML_ERROR_CODE	Out	DWORD	XML Fehler Information
ODK_ERROR_CODE	Out	DWORD	ODK Fehler Information

Der zeitliche Verlauf von REQ, BUSY, DONE und ERROR ist im Kapitel 4.3.1.

4.5.1 Datentypen

Der UDT2 XML_ITEM_TYP beschreibt die Daten auf die man in einer XML Datei zugreifen will.

Tabelle 4-5 UDT2 XML_ITEM_TYP Struktur:

Parameter	In/Out	Typ	Komentar
XPath	In	STRING[254]	XPath Ausdruck
dataType	In	BYTE	Datentyp siehe Tabelle 4-4
quantity	In	INT	Anzahl Werte (Array>1)
dbNumber	In	INT	DB Nummer (Wenn memoryArea Typ ein DB ist)
memoryArea	In	BYTE	Typ des Speicherbereichs: DB, M, E, A,... siehe Tabelle 4-5
areaOffset	In	INT	Speicherbereich Byte Offset
bitNumber	In	BYTE	Speicherbereich Bit Offset

Folgende Datentypen (Parameter „**dataType**“) des Speicherbereichs werden unterstützt:

Tabelle 4-6 mögliche dataType:

Hex	Datentyp
0x01	Bit
0x02	Byte
0x03	Char
0x04	Word
0x05	Integer
0x06	Double Word
0x07	Double Integer
0x08	Real
0x09	Date
0x0A	Tod
0x0B	Time
0x13	S7-String

Als Speicherbereich (Parameter „**memoryArea**“) kommen folgende Typen in Frage:

Tabelle 4-7 mögliche memoryArea:

Hex	Speichertyp
0x80	(P) Peripheral
0x81	(E) Process image
0x82	(A) Process image
0x83	(M) Flag
0x84	(DB) Data Block
0x01	(DS) Data Set
0x03	(SD) AS200:SYS-AREA
0x04	(S) AS200:stage bits
0x05	(SF) AS200:special flag
0x06	(AI) AS200:analog input
0x07	(AQ) AS200:analog output
0x08	(SZL) System State List

4.6 FBXMLSet

Mit diesem FB werden die XML Daten mit den Werten aus der SPS geschrieben.

Die Selektion, welche XML Daten gelesen werden sollen, erfolgt über ein XPath-Ausdruck. Die Selektion mehrerer gleichartiger Elemente ist möglich. Ausgangsseitig wird in diesem Fall ein Array zurückgegeben.

Tabelle 4-8 Parameter des FBXMLGet

Parameter	In/Output	Typ	Beschreibung
ODK_REF	In	WORD	Referenz zum Treiber (siehe FBFileInit)
ID	In	INT	Ident für diese Datei 0..31
XML_ITEM	in	XML_ITEM_TYP	Beschreibung der XML_ITEMS siehe Tabelle 4-3
RESET	In	BOOL	Reset interner Variablen (muss vor erster Benutzung einmalig gesetzt sein)
REQ	In	BOOL	Positive Flanke öffnet die Datei
BUSY	Out	BOOL	Auftrag läuft

Parameter	In/Out	Typ	Beschreibung
DONE	Out	BOOL	Auftrag fertig ohne Fehler
ERROR	Out	BOOL	Fehler aufgetreten
XML_ERROR_CODE	Out	DWORD	XML Fehler Information
ODK_ERROR_CODE	Out	DWORD	ODK Fehler Information

Der zeitliche Verlauf von REQ, BUSY, DONE und ERROR ist im Kapitel 4.3.1.

5 Fehlermeldungen

Der WinAC XMLDataAccess kann verschiedene Klassen von Fehlermeldungen liefern:

- Code im FB-Ausgang ODK_ERROR_CODE gemäß WinAC-ODK (siehe Kapitel 5.1 in diesem Dokument)
- Spezielle Fehlermeldungen des WinAC XMLDataAccess (siehe Kapitel 5.2 in diesem Dokument)

5.1 Fehlermeldungen vom WinAC ODK

Der Treiber wurde mit dem WinAC ODK (Open Development Kit) entwickelt. Das ODK kann ebenfalls Fehlercodes generieren, die im ODK_ERROR_CODE der FBs zurückgegeben werden.

Tabelle 5-1 WinAC ODK System-Fehlermeldungen

ODK Code (HEX)	Description
0	Success
8001	An exception occurred.
8002	Input: the ANY pointer is invalid.
8003	Input: the ANY pointer range is invalid.
8004	Output: the ANY pointer is invalid.
8005	Output: the ANY pointer range is invalid.
8006	More bytes were written into the output buffer by the extension object than were allocated.
8007	ODK system has not been initialized: no previous call to SFB65001 (CREA_COM).
8008	The supplied handle value does not correspond to a valid extension object.
8009	More bytes were written into the input buffer by the extension object than were allocated.
807F	An internal error occurred.
80C3	Maximum number (32) of parallel jobs/instances exceeded.
8102	The call to CLSIDFromProgID failed.
8103	The call to CoInitializeEx failed.
8104	The call to CoCreateInstance failed.
8105	The library failed to load.
8106	A Windows response timeout occurred.
8107	Controller is in an invalid state for scheduling an OB.
8108	Schedule information for OB is invalid.
8109	Instance ID for SFB65001 call is invalid.
810A	Controller could not load proxy DLL.
810B	The WinAC controller could not create or initialize shared memory area.
810C	Attempt to access unavailable option occurred.
8201	The Execute command index could not be found

ODK Code (HEX)	Description
8250	No more available positions in the job list
8252	The count is invalid
8253	A data type of an item in the list is invalid
8254	The count specified is invalid
8255	A memory area of an item in the list is invalid
8256	A DB number of an item in the list is invalid
8257	A bit number of an item in the list is invalid
8258	A pBuff of an item in the list is invalid
8259	A data quantity is invalid
825A	The area offset parameter is invalid for this type
825B	The frequency value is invalid
825C	The callback pointer is invalid
825D	The job ID pointer is invalid
825E	The job ID is invalid
825F	Job could not be completed because address is incorrect
8260	Job could not be completed because of protection level
8261	Job could not be completed because of hardware issues
8301	Invalid Thread Execution Priority
8401	Invalid Asynchronous Event
8402	Asynchronous Processor Queue is empty
8403	Asynchronous Processor Queue is full

5.2 Spezielle Fehlermeldungen des WinAC XMLDataAccess

Neben dem allgemeinen Fehlerbit der FBs wird im XML_ERROR_CODE ein spezieller Fehlercode geliefert, der die Ursache genauer beschreibt.

Hinweis Eine ganze Reihe von Fehlern liefern zusätzliche Informationen im Instanz-DB (tOdklf.dwErrInfo1 .. 4)

Tabelle 5-2 Fehlernummern vom WinAC XMLDataAccess

```
//No Error:  
0x80000000 // kein Fehler aufgetreten  
  
//XML Additional Errors:  
0x80000002 //Error on save data  
0x80000003 //XPath syntax error  
0x80000004 //No Items in XMLGet function  
0x80000005 //Data counts is wrong.  
0x80000006 //No items found in XMLSet function  
0x80000007 //XML Items count is wrong.  
0x80000040 // XML Doc is not open  
0x80000050 // Error on ODK input buffer.  
0x80000051 // Error on ODK async. event.  
0x80000052 // Error on ODK output buffer.
```

6 Literaturhinweis

6.1 Literaturangaben

Diese Liste ist keinesfalls vollständig und spiegelt nur eine Auswahl an geeigneter Literatur wieder.

Tabelle 6-1

	Themengebiet	Titel
/1/	STEP7	Automatisieren mit STEP7 in AWL und SCL Hans Berger Publicis MCD Verlag ISBN 3-89578-113-4
/2/	WinAC	Windows Automation Center RTX – WinAC RTX 2010 Betriebsanleitung
/3/		

6.2 Internet-Link-Angaben

Diese Liste ist keinesfalls vollständig und spiegelt nur eine Auswahl an geeigneten Informationen wieder.

Tabelle 6-2

	Themengebiet	Titel
\1\	Referenz auf den Beitrag	http://support.automation.siemens.com/WW/view/de/78028682
\2\	Siemens Industry Online Support	http://support.automation.siemens.com
\3\	WinAC RTX 2010 Betriebsanleitung	http://support.automation.siemens.com/WW/view/de/43715176
\4\		

7 Historie

Tabelle 7-1 Versions-Historie

Version	Datum	Modifizierungen
V1.0	08.2013	Erste Version