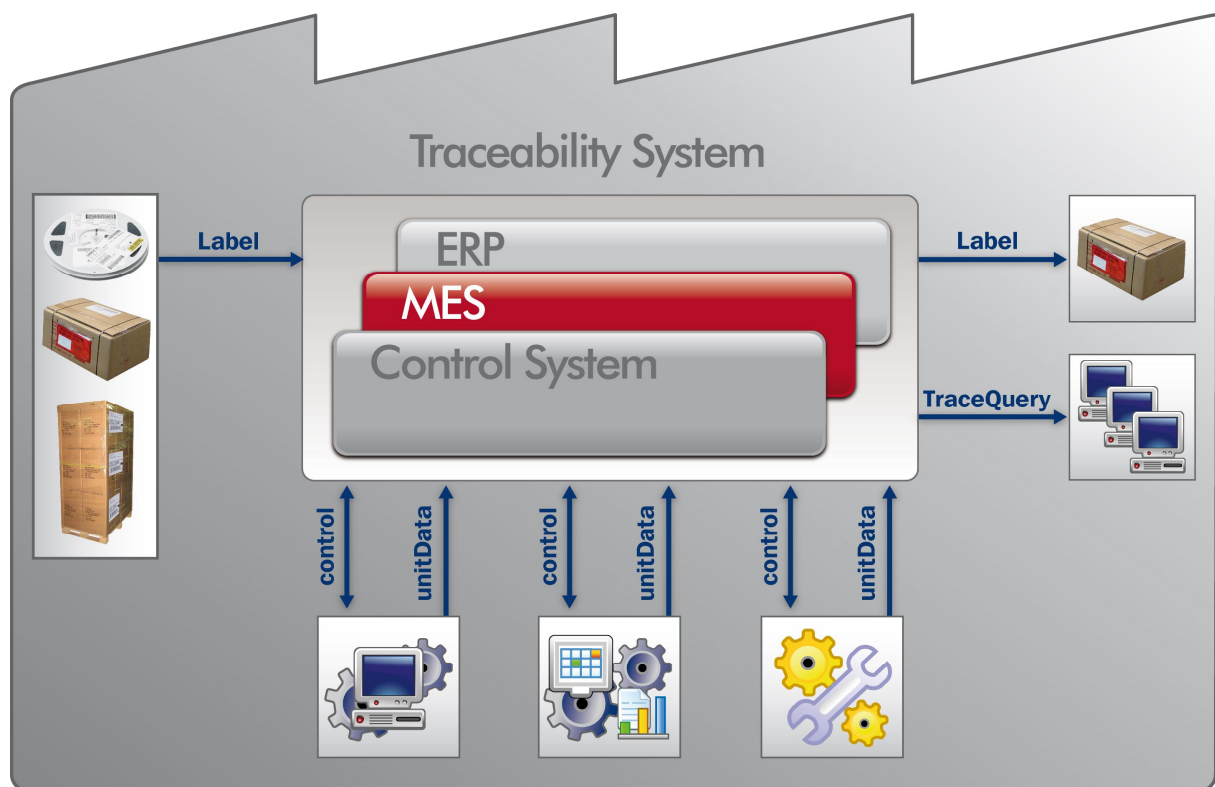


## Identifikation und Traceability in der Elektro- und Elektronikindustrie



## ZVEI Schnittstellen zum Shopfloor Übergabeprotokolle

Version 1.1.0

## Vorwort

Mit der Arbeit an dem ZVEI Leitfaden für die gesamte Liefer- und Wertschöpfungskette (siehe LIT-1 "Leitfaden für Identifikation und Traceability") wurde in der Arbeitsgruppe III Technology auch eine Schnittstelle zum Shopfloor für die Anbindung von Maschinen, Geräten und Arbeitsplätzen entworfen.

Ein Ziel war eine prozessübergreifende Standardisierung dieser Schnittstelle.

Als Ergebnis entstanden zwei xml-basierende Schnittstellen, die frei zur Verfügung stehen und vom ZVEI zur Anbindung des Shopfloor empfohlen werden.

- **control** für die Übertragung von Daten (Anfragen und Rückmeldungen) bei der Prozesskontrolle (advanced process control) während der Bearbeitung eines Produktes
- **unitData** für die Übertragung von Daten zur Bearbeitung eines Produktes

## Historie / Änderungen

In LIT-2 "ZVEI-Interfaces-ChangeHistory" ist die Historie der Änderungen der Schnittstellen **control** und **unitData** beschrieben.



ZVEI - Zentralverband Elektrotechnik-  
und Elektronikindustrie e.V.

Fachverband

Electronic Components and Systems

Lyoner Straße 9

60528 Frankfurt am Main

Fon: 069 6302 – 276

Fax: 069 6302 – 407

Mail: [zvei-be@zvei.org](mailto:zvei-be@zvei.org)

[www.zvei-traceability.de](http://www.zvei-traceability.de)

# Inhaltsverzeichnis

<b>Vorwort</b>	<b>2</b>
<b>Historie / Änderungen</b>	<b>2</b>
<b>Inhaltsverzeichnis</b>	<b>ii</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Verwendete Symbole	1
1.2 Legende	1
<b>2 Übergabeprotokolle</b>	<b>2</b>
2.1 Datei	2
2.1.1 Dateinamenkonvention	2
2.1.1.1 [Typ_]	2
2.1.1.2 Bezug	2
2.1.1.3 Zeitstempel	3
2.1.1.4 [_Zähler]	3
2.1.1.5 Beispiele für Dateinamen	3
2.1.2 Dateihandling	4
2.2 TCP/IP	6
2.2.1 Ablauf	6
2.2.2 Überwachung der TCP-Verbindung	7
2.2.2.1 Timeout	7
2.2.2.2 Keepalive	7
2.2.2.3 Verbindungsanzahl pro IP-Adresse	7
<b>3 Anhang</b>	<b>8</b>
3.1 Verzeichnis relevanter Dokumente	8
3.2 Verzeichnis der Abbildungen:	8
3.3 Verzeichnis relevanter Begriffe und Abkürzungen	8

# 1 Einleitung

In diesem Dokument sind die Übergabeprotokolle der ZVEI Standardschnittstellen zur Übertragung von XML-Strukturen beschrieben.

## 1.1 Verwendete Symbole

In dieser Dokumentation werden drei unterschiedliche Symbole verwendet, die wichtige Inhalte hervorheben sollen.



### **Achtung!**

Dieses Symbol weist auf wichtige Informationen, die unbedingt beachtet werden müssen.



### **Hinweis!**

Dieses Symbol weist auf erklärende Informationen hin.



### **Tipp!**

Dieses Symbol hält Tipps für schnellere oder effizientere Lösungen bereit.

## 1.2 Legende

[ Knoten/Attribut ]      Eckige Klammern: → optionaler Knoten/Attribut

< Knoten/Attribut >      Spitze Klammern: → alternativer Knoten/Attribut



### **Hinweis!**

Wenn ein Attribut erforderlich ist (nicht optional) , muss auch der Value belegt sein (kein Leerstring).

Wenn ein Attribut nicht erforderlich ist (optional), aber mit dem Value = "" (Leerstring) vorhanden ist, wird das Attribut ignoriert. Das Attribut wird dann so behandelt, als wäre es gar nicht vorhanden.

## 2 Übergabeprotokolle

Zur Übertragung der XML-Strukturen stehen verschiedene Übergabeprotokolle zur Verfügung.

### 2.1 Datei

In diesem Kapitel sind die Voraussetzungen und Einschränkungen für die Übergabe von XML-Strukturen mit einer Dateischnittstelle beschrieben.

#### 2.1.1 Dateinamenkonvention

Der Dateiname jeder XML-Datei muss eindeutig sein. Um die Festplattenzugriffe gering zu halten, soll der Empfänger bereits aus dem Dateinamen wichtige, ausgewählte Informationen erhalten können, ohne die Datei öffnen zu müssen.

Der Dateiname setzt sich deshalb wie folgt zusammen:

**[Typ\_]Bezug\_Zeitstempel[\_Zähler].xml**

##### 2.1.1.1 [Typ\_]

Der optionale Typ im Dateinamen kann genutzt werden, um die Art der übertragenden Daten zu kennzeichnen.

Beispiele:

'R\_' für Request bei bidirektionaler Verbindung

'A\_' für Answer bei bidirektionaler Verbindung

'PD\_' für Processdata



#### **Achtung!**

Wird für unterschiedliche Typen von XML-Strukturen (z.B. verschiedene ZVEI Standardschnittstellen oder bidirektionale Verbindungen mit Anfrage und Antwort) das gleiche Übergabeverzeichnis für den Dateiaustausch verwendet, so soll eine Typkennung im Dateinamen enthalten sein.

##### 2.1.1.2 Bezug

Der Dateiname soll einen Bezug zu den Dateninhalten der Datei haben. Dies kann zum Beispiel eine Serien-, Auftrags-, Maschinen- oder Sensornummer sein.

### 2.1.1.3 Zeitstempel

Die Eindeutigkeit eines Dateinamens wird mittels eines Zeitstempel und optional zusätzlich durch einen Zähler gewährleistet. Dabei ist unbedingt das Format des Zeitstempels zu berücksichtigen. Es dürfen nicht mehrere Dateien innerhalb der kleinsten Zeiteinheit dieses Formates geschrieben werden.

#### Beispiele für Zeitstempel:

- YYYYMMDD\_hhmmss (JahrMonatTag\_StundeMinuteSekunde)
- YYMMDD\_hhmmssSSS (JahrMonatTag\_StundeMinuteSekundeMillisekunden)



#### Achtung!

Wenn Dateien länger als eine Stunde nicht abgearbeitet werden können und auch am Wochenende die Produktion laufen kann, so ist unbedingt die jährliche Umstellung von der Sommerzeit auf die Winterzeit zu beachten. Diese Anforderung kann vom Sender mit einem Zähler im Dateinamen oder einem Prüfen noch vorhandener Dateien gelöst werden.

### 2.1.1.4 [\_Zähler]

Optional kann im Dateinamen ein Zähler enthalten sein.



#### Achtung!

Können innerhalb der kleinsten Zeiteinheit des Zeitstempels mehrere Dateien geschrieben werden, ist die Eindeutigkeit der Dateien nicht mehr gegeben. Dann muss die Eindeutigkeit durch verwenden eines Zählers erfolgen.

### 2.1.1.5 Beispiele für Dateinamen

Einige Beispiele für Dateinamen sind:

- Bezug\_Zeitstempel.xml  
Seriennummer\_YYYYMMDD\_hhmmss.xml  
Beispiel: 4711001\_20090502\_080000.xml
- Typ\_Bezug\_Zeitstempel.xml  
Anfrage\_Auftragsnummer\_YYYYMMDD\_hhmmss.xml  
Beispiel: R\_4712001\_20090502\_080000.xml
- Typ\_Bezug\_Zeitstempel\_Zähler.xml  
Processdaten\_Sensornummer\_YYYYMMDD\_hhmmss\_Zähler.xml  
Beispiel: PD\_4713001\_20090502\_080000\_12.xml



#### Hinweis!

Die Namenskonvention wurde bei den allgemeinen Beispieldateien nicht verwendet, da anhand der Dateinamen die jeweiligen Schnittstellen und deren Version erkennbar sein sollen.

## 2.1.2 Dateihandling

Bei allen ZVEI Datei-Standardschnittstellen erfolgt mit der Übergabe der Datei auch der Wechsel der Zuständigkeiten:

- Der Sender einer Datei muss vor der Übergabe die Datei komplett schreiben und alle Datei-Kanäle schließen bzw. eventuelle Datei-Sperren freigeben.
- Der Sender darf nach erfolgter Übergabe die Datei nicht erneut öffnen bzw. sperren.
- Der Empfänger darf vor der Übergabe die Datei nicht öffnen bzw. sperren.
- Der Empfänger ist für die Archivierung und das Löschen der empfangenen Datei zuständig.

Die Übergabe einer Datei erfolgt durch Verschieben der Datei in ein Übergabeverzeichnis. Hierbei werden folgende Fälle unterstützt:

- 1) Sender und Empfänger befinden sich auf demselben Rechner und verwenden ein lokales Dateisystem zur Übergabe.
- 2) Sender und Empfänger befinden sich auf verschiedenen Rechnern und verwenden ein zum Sender lokales Dateisystem zur Übergabe.
- 3) Sender und Empfänger befinden sich auf verschiedenen Rechnern, benutzen aber ein zum Empfänger lokales Dateisystem zur Übergabe.

Befinden sich Sender und Empfänger auf unterschiedlichen Rechnern, so ist Fall 2) zu bevorzugen, da der Sender ansonsten bei Ausfall des Netzwerkes über ein separates Pufferkonzept, ein nicht mehr mögliches Wegschreiben der Dateien absichern muss.

Folgende Vorgehensweise muss eingehalten werden, um eine reibungslose Übergabe zu gewährleisten:

- a) Der Sender schreibt eine Datei zuerst in ein Arbeitsverzeichnis. Danach verschiebt er diese Datei in das Übergabeverzeichnis **im selben Dateisystem!** Nur so ist die Verschiebeoperation **atomar** und es kann nicht zu einem gegenseitigen Blockieren kommen!
- b) Erkennt der Empfänger eine Datei im Übergabeverzeichnis, so beginnt er deren Bearbeitung.

Handelt es sich um ein Sender lokales Übergabeverzeichnis (Fall 2), so ist folgende Vorgehensweisen zu nutzen:

- Die Originaldatei wird zu Beginn der Bearbeitung in ein Empfänger lokales Arbeitsverzeichnis verschoben, dort bearbeitet und nach erfolgreicher Abarbeitung archiviert oder gelöscht.

Die Originaldatei ist je nach Anforderungen vom **Empfänger** zu archivieren bzw. zu löschen. Das Löschen der Originaldatei aus dem Übergabeverzeichnis kann je nach Anforderung zu unterschiedlichen Zeitpunkten erfolgen:

- Bei asynchroner Abarbeitung (Keine Rückmeldung über die erfolgreiche Abarbeitung notwendig.) kann die Originaldatei bereits zu Beginn der Bearbeitung gelöscht werden. Dies ist speziell bei einem zum Empfänger entfernten Übergabeverzeichnis sinnvoll. (Ansonsten muss bei einem Netzwerkausfall überprüft werden, ob die Datei schon abgearbeitet wurde oder nicht.)
- Bei synchroner Abarbeitung (Eine Rückmeldung über die erfolgreiche Abarbeitung ist notwendig.) darf die Originaldatei erst nach erfolgreicher Abarbeitung gelöscht werden. Das Löschen signalisiert in diesem Fall das Ende der Bearbeitung.



**Achtung!**

Hinweis zur Gültigkeit von Daten:

Nach einem Netzwerkausfall oder einer Auszeit des Empfängers können Daten veraltet sein. Die Bedingung für das Verfallen von Daten und das dann notwendige Handling muss für den speziellen Fall eigens spezifiziert werden. Gleiches gilt für Überschreitung von Time-Outs bei einem Handshake-Mechanismus zwischen den Kommunikationspartnern.



**Achtung!**

Fehlerhandling:

Bis die Schnittstellendatei im Übergabeverzeichnis bereit liegt, ist ausschließlich der Sender für das Fehlerhandling verantwortlich.

Die Verantwortung für das Fehlerhandling vom Empfang der Datei aus dem Übergabeverzeichnis bis zu deren Abarbeitung liegt beim Empfänger.

Das Fehlerhandling sollte ein Loggen und Melden von Fehlern beinhalten.



**Achtung!**

Konfiguration:

Alle Kommunikations- und Archiv-Verzeichnisse müssen in den Systemen aller Kommunikationspartner konfigurierbar sein.



## 2.2 TCP/IP

In diesem Kapitel sind die Voraussetzungen und Einschränkungen für die Übergabe von XML-Strukturen mit einer mit einer TCP/IP-Schnittstelle beschrieben.

### 2.2.1 Ablauf

Für die Übergabe von XML-Strukturen mit einer TCP/IP-Schnittstelle müssen im Client die IP-Adresse des Servers und die Portnummer der Verbindung konfigurierbar sein.

Bei der Kommunikation werden die Daten als reiner Bytestream übertragen.

Das folgende Sequenzdiagramm zeigt einen einfachen Verlauf der Kommunikation zwischen einem Client und einem Host. Der Client versucht sich beim Host mit 'connect' über einen bekannten Port zu verbinden. Kommt die Verbindung zustande, sendet der Client über die neue Verbindung mit 'send' die Daten als Bytestream an den Host.

Der Host liest die Daten aus dem ByteStream und übergibt diese an einen XML-Parser, der die Daten an Hand der im ByteStream referenzierten Schema Datei validiert und anschließend entsprechend aufbereitet.



#### Hinweis!

Zur Validierung der Daten und zur Identifizierung des Typs des gesendeten Datenpakets muß im XML Stream ein gültiger Verweis als URL auf die definierende Schema Datei vorhanden sein. Siehe hierzu die Spec. der jeweiligen xml-Formate.

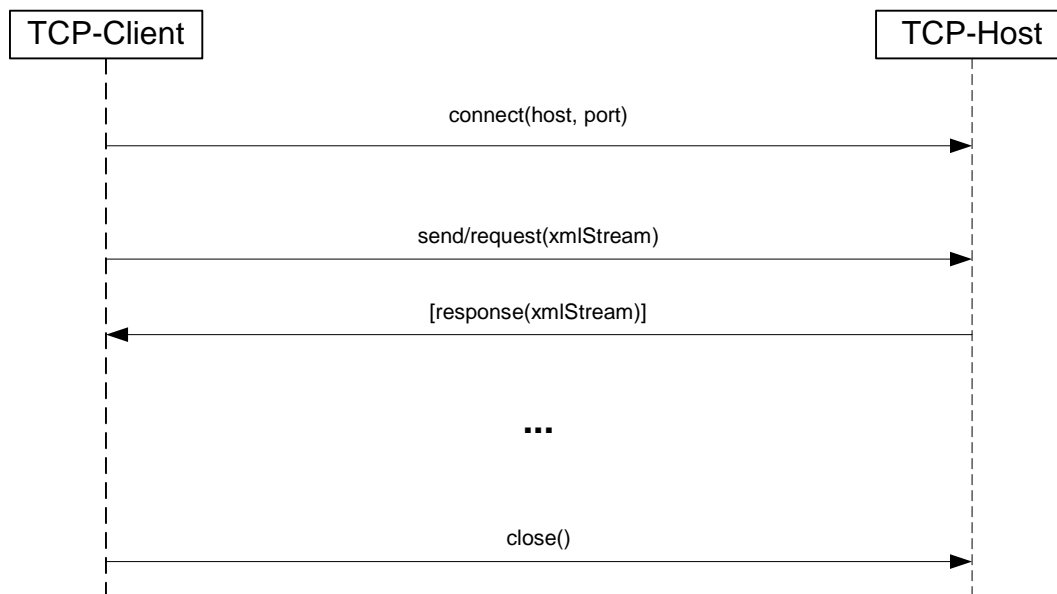


Abbildung 1: Typische Sequenz beim Übertragen einer XML-Struktur

## **2.2.2 Überwachung der TCP-Verbindung**

Bricht eine bestehende TCP-Verbindung zusammen, so wird dies nicht immer zeitnah vom Betriebssystem gemeldet. Für die Kommunikationspartner ist in einer solchen Situationen nicht erkenntlich, dass über eine Verbindung keine Daten mehr gesendet werden.

Um zu verhindern, dass inaktive Verbindungen auf lange Zeit offen gehalten werden, gibt es mehrere Möglichkeiten zur Erkennung dieser inaktiven Verbindungen.

### **2.2.2.1 Timeout**

Es kann eine maximale Zeitspanne konfiguriert werden, über die eine Verbindung offen gehalten wird, ohne dass Daten empfangen werden. Ist diese Zeitspanne überschritten, so wird die Verbindung automatisch geschlossen.

### **2.2.2.2 Keepalive**

Es kann in regelmäßigen Abständen eine Keepalive-Nachricht über die TCP-Verbindung gesendet werden, um zu testen, ob diese noch verfügbar ist. Tritt beim Senden der Keepalive-Nachricht ein Fehler auf, so wird die Verbindung geschlossen.

Inhalt und Sendezyklus der Keepalive-Nachricht soll konfigurierbar sein. Der Kommunikationspartner muss diese Keepalive-Nachricht ignorieren.

### **2.2.2.3 Verbindungsanzahl pro IP-Adresse**

In den meisten Anwendungsfällen gibt es nur eine aktive TCP-Verbindung zu einem TCP-Host pro Arbeitsplatz.

Um sicherzustellen, dass pro Arbeitsplatz nicht mehr Verbindungen verwendet werden, kann beim Öffnen einer Verbindung geprüft werden, ob es bereits eine bestehende Verbindungen gibt. In diesem Fall wird die alte Verbindungen beendet bevor eine neue Verbindung aufgebaut wird.

## 3 Anhang

### 3.1 Verzeichnis relevanter Dokumente

LIT-1 " <a href="#">Leitfaden für Identifikation und Traceability</a> "	ZVEI-Leitfaden für die gesamte Liefer- und Wertschöpfungskette
LIT-2 "ZVEI-Interfaces-ChangeHistory"	Dieses Dokument beschreibt die Historie der Änderungen der Schnittstellen <b>control</b> und <b>unitData</b> .

### 3.2 Verzeichnis der Abbildungen:

Abbildung 1: Typische Sequenz beim Übertragen einer XML-Struktur .....	6
--	---

### 3.3 Verzeichnis relevanter Begriffe und Abkürzungen

Begriff	Beschreibung
Sender	Kommunikationspartner, der Dateien zur Verfügung stellt
Empfänger	Kommunikationspartner, der Dateien entgegennimmt und weiterverarbeitet
Kommunikations-Verzeichnis	Dateiverzeichnis, über das die Kommunikationspartner Dateien mit Daten austauschen
Node	Knotenpunkt in der xml-Baumstruktur
Sub Node	Knotenpunkt in der xml-Baumstruktur unterhalb eines Node