



OpCon Xml Spezifikation V2.2

Änderungsindex

Version	Änderung	Geändert durch	Datum
V2.02	<ul style="list-style-type: none"> • Neuer WorkingCode GoldenDevice-Teil • Datentyp von result und workingCode in resHead auf short (INT) geändert • Datentypen in SPS-Datentypen umgewandelt 	Eiberle	24.11.2009
V2.01	<ul style="list-style-type: none"> • Erweiterung resHead um machineID 	Eiberle	16.04.2009
V2.00	<ul style="list-style-type: none"> • Einfügen der Anmerkungen von AE-MES Team • Erweiterungen in plcToolChangeXXX und plcMaterialChangeXXX • Beschreibung Vorgabewerte 	Eiberle	13.03.2009
V2.00 (beta 2)	<ul style="list-style-type: none"> • Serverseitiges TraceLevel entfernt • Hinweis Inhalt timeStamp • Neues Attribut contentType in header-Element • Erweiterung Teil II: <structs>-Element 	Eiberle	06.03.2009
V2.00 (beta)	<ul style="list-style-type: none"> • Überarbeitung für Anforderungen an MES-Maschinenschnittstelle 	Eiberle	21.02.2009
V1.05	<ul style="list-style-type: none"> • Fehler in der Definition von Standard-Arrays behoben • Falscher Eintrag in resHead-Definition richtig gestellt (cycleTimePrev) 	Eiberle	02.03.2007
V1.04	<ul style="list-style-type: none"> • Neues Master-Dokument • Review und kleinere Anpassungen 	Eiberle	26.02.2007
V01.03	<ul style="list-style-type: none"> • dataUploadRequired und partProcessed für LLD erweitert 	Eiberle	04.08.2006
V01.02	<ul style="list-style-type: none"> • Beispiele überarbeitet • productVersion Elemente aus <body> entfernt • Antwort bei partReceived (Anwendung auf LLD geändert) 	Eiberle	10.06.2005
V01.01	<ul style="list-style-type: none"> • dataUploadRequired und partProcessed für LLD erweitert 	Eiberle	07.06.2005
V01.00	Erste freigegebene Version	Eiberle	12.05.2005
V00.09	Überarbeitung nach Rücksprache mit Kunde	Eiberle	29.04.2005
V00.01	Erstauflage	Eiberle	05.04.2005

Inhaltsverzeichnis

1	Einführung	4
1.1	Allgemein	4
1.2	Konventionen	4
1.3	Datentypen	5
1.4	Beispiele	5
2	Teil 1: Kern-Spezifikation	6
2.1	Transportprotokoll	6
2.2	Allgemeiner Aufbau der XML-Nachricht	6
2.2.1	Beschreibung der Elemente	7
2.3	Abbildung der Ereignisse	14
2.3.1	Abbildung der Einrichtungs-Ereignisse	14
2.3.2	Abbildung der Prozess-Ereignisse	25
2.3.3	Abbildung der Daten-Ereignisse	27
	Kommunikationsablauf	29
3	Teil 2: Erweiterte Elemente	31
3.1	<items>-Element	31
3.1.1	<item>-Element	31
3.2	<resHead>-Element	32
3.3	Benutzerdefinierte Elemente	33
3.3.1	<structs>-Element	33
3.4	Felder	34
3.4.1	<arrays>-Element (Standard-Felder)	34
3.4.2	<structArrays>-Element (Struktur Felder)	34
3.4.3	Benutzerdefinierte Felder	37
	Anhang A: Vorgabewerte	38

1 Einführung

1.1 Allgemein

OpCon Xml ist ein Standard zum Austausch von Daten zwischen Fertigungseinrichtungen und OpCon MES. Die von OpCon DirectDataLink vorgegebenen Ereignisse werden dabei als entsprechende XML-Telegramme umgesetzt.

OpCon Xml wurde entwickelt um eine möglichst breite Plattform an Fertigungseinrichtungen abzudecken. Die hier definierten Elemente müssen dabei nicht von allen Plattformen in vollem Umfang zwingend unterstützt werden. So können einfache Clients (z. B. SPS) nur die Kernteile der Spezifikation unterstützen, während andere, mächtigere Clients (z. B. PC-basierende Systeme), den kompletten Standard implementieren. Um OpCon Xml in einem möglichst breiten Spektrum an Applikationen einsetzen zu können, wurde der Standard in drei Bereiche aufgeteilt:

1. Kern-Spezifikation

Dieser Teil definiert die Teile welche von allen Clients zwingend implementiert werden müssen. Die in diesem Teil enthaltenen Definitionen müssen immer eingehalten werden.

2. Erweiterte Elemente

In diesem Teil sind die Definitionen enthalten, die von einem Client implementiert werden können. Je nach Fähigkeit des Clients können die hier festgelegten Definitionen implementiert werden. Implementiert ein Client eine Definition, so muss er diese jedoch vollständig implementieren. Es ist nicht zulässig nur Teile einer Definition zu implementieren.

3. Bereichsspezifische Elemente

Verschiedene Fertigungsbereiche haben verschiedene Anforderungen an die Kommunikation. In diesem Teil können die bereichsspezifischen Definition hinterlegt werden (z. B. spezifische Definitionen die für den gesamten AE-Bereich gültig sind).

Das vorliegende Dokument enthält Teil 1 (Kern-Spezifikation) und Teil 2 (Erweiterte Elemente). Teil 3 wird bereichsspezifisch in einem oder mehreren weiteren Dokumenten beschrieben.

1.2 Konventionen

Schriftarten

Monospace Alle Texte in Monospace bezeichnen Ausschnitte aus Konfigurationsdateien oder Quellcode

fett Alle fettgedruckten Wörter bezeichnen wichtige Begriffe

kursiv Begriffe die zum ersten Mal eingeführt werden, werden kursiv dargestellt. Die Definition kursiver Begriffe ist auch im Glossar hinterlegt.

Rahmen Definition werden mit einem Rahmen versehen

Symbole

Definitionen



Hinweise



Tipps



Verweis auf weiterführende Dokumentation

Notation

- . Trennzeichen zwischen XML-Elementen
- @ Trennzeichen zwischen XML-Element und XML-Attribut

1.3 Datentypen

In der folgenden Spezifikation werden die folgenden Datentypen verwendet:

SPS Datentyp	OpCon Datentyp	Hochsprache Datentyp	Kommentar
UDINT	19	uint	32-bit integer, vorzeichenlos
DINT	3	int	32-bit integer, vorzeichenbehaftet
INT	2	short	16-bit integer, vorzeichenbehaftet
STRING	8	char[80]	Zeichenkette, max. Länge 80 Zeichen
STRING(x)	8	char[x]	Zeichenkette, max. Länge x Zeichen
BOOL	11	bool	bool, kann die Literale 'true' oder 'false' enthalten. Groß-/Kleinschreibung muss beachtet werden.
LREAL	5	double	8-Byte real, Dezimaltrennzeichen ist ein Punkt, e-Darstellung ist erlaubt und bevorzugt.
REAL	4	float	4 Byte real, Dezimaltrennzeichen ist ein Punkt, e-Darstellung ist erlaubt und bevorzugt.

1.4 Beispiele

Diese Spezifikation enthält eine Reihe von Beispielen. Diese Beispiele dienen der Veranschaulichung der jeweiligen Definitionen. Sie haben keinen unmittelbar bindenden Charakter. Insbesondere können in den Beispielen optionale Elemente benutzt sein, um verschiedene Aspekte der Definition besser herauszustellen. Es ist somit zulässig, dass Implementierungen sich von den in den Beispielen gezeigten XML-Telegrammen unterscheiden können, sofern die Implementierung keine der in diesem Dokument beschriebenen Definitionen verletzt.

**Hinweis**

Die Beispiele in diesem Dokument dienen zur Veranschaulichung. Sie haben keinen unmittelbar bindenden Charakter.

2 Teil 1: Kern-Spezifikation

2.1 Transportprotokoll

Die Übertragung der Nachrichten erfolgt über TCP/IP. Dabei wird der eigentlichen Nachricht ein 4-Byte großer Nachrichtenkopf vorangestellt, welcher die Gesamtgröße der zu übertragenden Nachricht enthält. Die Gesamtgröße setzt sich auch:

- Größe der Nachricht in Bytes
- Größe des Nachrichtenkopfs (4 Byte)

zusammen.

Die Größe muss in Network byte order (Big Endian) übertragen werden. Durch die Verwendung von TCP/IP ist keine Größenbeschränkung der Nachricht vorgesehen.

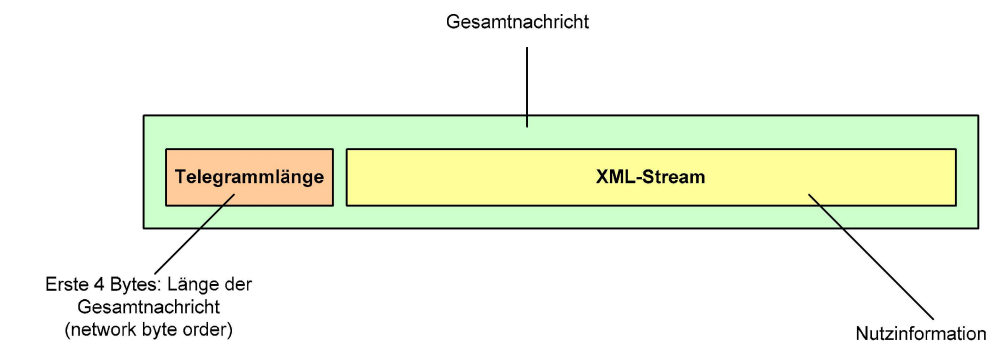


Abbildung 1 Aufbau Transportprotokoll

2.2 Allgemeiner Aufbau der XML-Nachricht

Die XML-Nachricht hat den folgenden strukturellen Aufbau:

```
<?xml version="1.0" encoding="UTF-8"?>

<root>
  <header/> <!-- zwingend -->
  <event/> <!-- zwingend, Aufbau ereignisspezifisch -->
  <body/> <!-- zwingend, Aufbau applikationsspezifisch -->
</root>
```

Abbildung 2 Aufbau XML-Nachricht

Dieser Aufbau ist unabhängig von der Art der Information die übertragen werden soll. Der Aufbau wird sowohl für Nachrichten verwendet, die an MES gesendet werden, wie auch für Nachrichten die von MES an die einzelnen Stationen gesendet werden. Die Kind-Elemente innerhalb der Elemente `<event>` und `<body>` können unterschiedlich sein. Das Element `<header>` besitzt einen fixen Aufbau.



Hinweis

MES garantiert nicht die korrekte Verarbeitung von Zeichen außerhalb des US-ASCII-Zeichensatzes auf der Clientseite. Dies muss applikationsspezifisch immer verifiziert werden und ist nicht Bestandteil des Umfangs dieses Projekts.



Hinweis

Die OpCon SPS-Bibliotheken unterstützen nur Zeichenketten in US-ASCII.

2.2.1 Beschreibung der Elemente

2.2.1.1 <root>-Element

Das Element `<root>` ist das Wurzelement der XML-Nachricht. Es dient als Container für die Nutzinformationen.

Element	Angabe	Kommentar
<code><header></code>	zwingend	Identifikation Nachricht Detaillierte Beschreibung siehe Abschnitt 2.2.1.2.
<code><event></code>	zwingend	Enthält die ereignisspezifischen Kind-Elemente. Detaillierte Beschreibung siehe Abschnitt 2.2.1.4.
<code><body></code>	optional	Enthält die applikationsspezifischen Kind-Elemente. Detaillierte Beschreibung siehe Abschnitt 2.2.1.7.

Attribute

-

2.2.1.2 <header>-Element

Das Element `<header>` beinhaltet die Informationen zur Nachricht an sich. Es enthält alle Informationen die für die eindeutige Identifikation der Nachricht notwendig sind.

Attribute

Attribut	Angabe	Datentyp	Kommentar
eventId	zwingend	UDINT	Eindeutige Identifikation des Ereignisses. Diese Identifikation kann von Stationen verwendet werden, wenn Antworten auf Ereignisse ausgewertet werden sollen (z. B. <code>dataDownloadRequired</code>)
eventName	zwingend	STRING	Name des Ereignisses. Anhand dieses Namens wird das Ereignis identifiziert. Dieser Name legt zum einen die weitere Interpretation der Nachricht fest. Zum anderen wird durch diesen Namen die notwendige Verarbeitung des Ereignisses durch MES definiert.
version	zwingend	STRING(16)	Version des Nachrichtenprotokolls. Die Versionsnummer setzt sich aus Haupt- und Unterversion zusammen. Aufbau: Hauptversion.Unterversion, also z.B. "2.0" Versionsnummern werden bei PA-ATMO verwaltet und dokumentiert.
eventSwitch	optional	DINT	Dient zur Steuerung der Auswahl der richtigen Verarbeitung, wenn mehrere Verarbeitungen für das Ereignis möglich sind (z. B. bei <code>dataUploadRequired</code> , <code>dataDownloadRequired</code> ,...). Damit kann für ein Ereignis unterschiedliche Verarbeitungen hinterlegt werden, sprich es kann von der Station noch eine Fallunterscheidung vorgenommen werden (z. B. <code>dataDownloadRequired: eventSwitch=1</code> -> Stationsdaten werden benötigt, <code>eventSwitch=2</code> -> Systemdaten werden benötigt, ... Wird -1 angegeben, so wird diese Information ignoriert und es wird keine Unterscheidung vorgenommen.
timeStamp	optional	STRING	Zeitstempel des Ereignisses. Definiert den Zeitpunkt des Auftretens des Ereignisses. Der

			Zeitstempel muss gemäß W3C aufgebaut sein. Kann der Ereignisort diese Information nicht bereitstellen, so wird der Zeitpunkt des Empfangs der Nachricht als Zeitstempel von MES verwendet.
user	optional	STRING	Kennung (z. B. NT-Login) des Benutzers
pwd	optional	STRING	Passwort (verschlüsselt) des Benutzers. Der Verschlüsselungsalgorithmus muss dem verschlüsselten Passwort vorangestellt werden. Getrennt werden die beiden Informationen durch einen ':' [Algorithmus][:][Verschlüsseltes Passwort] Siehe auch: Abschnitt 2.3.1.21
contentType	optional	DINT	Definiert in welchem Format der Client die Antwort erwartet. Über den Inhalt definiert der Client die Elemente die verwendet werden können. Default: contentType=0

contentType Belegung

Die folgende Tabelle zeigt die Interpretation des contentTypes. Je nach gesendetem Wert können die angegebenen Elemente in einer Nachricht (Anfrage bzw. Antwort) enthalten sein.

contentType	items	arrays	benutzerdef. Elemente	benutzerdef. arrays	structs	structArrays	trace
0	x	x	x	x			
1	x	x	x	x			x
2	x	x			x	x	
3	x	x			x	x	x

**Hinweis**

timeStamp enthält immer die lokale Zeit und daran angehängt den Offset zu UTC-Zeit. Dieser Offset ist abhängig, ob Sommer- oder Winterzeit herrscht. Sprich für die Zeitzone Rom ist der Offset während der Winterzeit +1 und während der Sommerzeit +2.

Unterelemente

Element	Angabe	Kommentar
<location>	zwingend	Eindeutige Identifikation des Ereignisorts. Detaillierte Beschreibung in Abschnitt 2.2.1.3.

**Beispiel**

Das folgende <header>-Element beschreibt das Telegram für das Ereignis plcOperationModeChanged.

```
<header eventId="1"
  eventName="plcOperationModeChanged"
  version="2.0"
  eventSwitch="-1"
  timeStamp="2009-02-20T13:00:00.001+02:00"
/>
```


2.2.1.3 <location>-Element

Das <location>-Element dient zur Angabe des Ereignisorts für die Nachricht. Der Ereignisort setzt sich immer auch den folgenden Informationen zusammen:

- Logische Ortsangabe
- Auslösende Applikation

Unterelemente

-

Attribute

Attribut	Angabe	Datentyp	Kommentar
lineNo	zwingend	UDINT	Liniennummer Die Liniennummer muss eine Zahl zwischen 1..9999 sein.
statNo	zwingend	UDINT	Stationsnummer Die Stationsnummer muss eine Zahl zwischen 1..9999 sein.
statIdx	zwingend	UDINT	Stationsindex Der Stationsindex muss eine Zahl zwischen 1..9999 sein.
fuNo	optional	UDINT	Funktionseinheit-Index Der Index für eine Funktionseinheit muss eine Zahl zwischen 0..8 sein.
workPos	optional	UDINT	Bearbeitungsposition Die Bearbeitungsposition muss eine Zahl zwischen 0..9999 sein.
toolPos	optional	UDINT	Werkzeugposition Die Werkzeugposition muss eine Zahl zwischen 0..9999 sein.
processNo	optional	UDINT	Prozessnummer
processName	optional	STRING	Name des Prozesses
application	zwingend	STRING	Name der Applikation die das Ereignis auslöst.



Beispiel

Das folgende <location>-Element beschreibt den Ereignisort für den Prozess 'Flash EPROM'. Der Prozess hat die Prozess-ID 10 und existiert innerhalb der Linie 1 nur einmal. Das Ereignis an sich wird durch das SPS-Programm ausgelöst. Damit ergeben sich die folgenden Daten für den Ereignisort:

Liniennummer: 1
 Prozessnummer: 10
 Stationsindex: 1
 Funktionseinheit: 1
 Bearbeitungsposition: 1
 Werkzeugposition: 1
 Prozessnummer: 10
 Prozess: AOI
 Applikation: PLC

```

<location lineNo="1"
  statNo="10"
  statIdx="1"
  fuNo="1"
  workPos="1"
  toolPos="1"
  processNo="10"
  processName="AOI"
  application="PLC"
/>
  
```

2.2.1.4 <event>-Element

In diesem Element werden alle ereignisspezifischen Informationen abgelegt. Es besteht aus einer Reihe von optionalen Unterelementen, die jeweils eine spezifischen Ereignis zugeordnet sind.

Attribute

-

Unterelemente

Element	Angabe	Kommentar
<plcChangeOverStarted>	optional	Beinhaltet die Grundinformation für das Einrichtungs-Ereignis 'Beginn Umrüsten' . Detaillierte Beschreibung in Abschnitt 2.3.1.1.
<plcChangeOver>	optional	Beinhaltet die Grundinformation für das Einrichtungs-Ereignis 'Umgerüstet' . Detaillierte Beschreibung in Abschnitt 2.3.1.2.
<plcOperationModeChanged>	optional	Beinhaltet die Grundinformation für das Einrichtungs-Ereignis 'Betriebsart gewechselt' . Detaillierte Beschreibung in Abschnitt 2.3.1.3.
<plcEventOn>	optional	Veraltet. Wird in der aktuellen Spezifikation nicht mehr unterstützt.
<plcEventOff>	optional	Veraltet. Wird in der aktuellen Spezifikation nicht mehr unterstützt.
<plcSystemStarted>	optional	Beinhaltet die Grundinformation für das Einrichtungs-Ereignis 'Station/Einrichtung eingeschaltet/aktiviert' . Detaillierte Beschreibung in Abschnitt 2.3.1.6.
<plcStationSwitchedOff>	optional	Beinhaltet die Grundinformation für das Einrichtungs-Ereignis 'Station/Einrichtung ausgeschaltet/deaktiviert' . Detaillierte Beschreibung in Abschnitt 2.3.1.7.
<plcError>	optional	Beinhaltet die Grundinformation für das Einrichtungs-Ereignis 'Fehler aufgetreten/quittiert' . Detaillierte Beschreibung in Abschnitt 2.3.1.8.
<plcPartsMissingStarted>	optional	Beinhaltet die Grundinformation für das Einrichtungs-Ereignis 'Teilemangel erkannt' . Detaillierte Beschreibung in Abschnitt 2.3.1.9.
<plcPartsMissing>	optional	Beinhaltet die Grundinformation für das Einrichtungs-Ereignis 'Teilemangel' . Detaillierte Beschreibung in Abschnitt 2.3.1.10.
<plcJamStarted>	optional	Beinhaltet die Grundinformation für das Einrichtungs-Ereignis 'Stau erkannt' . Detaillierte Beschreibung in Abschnitt 2.3.1.11.
<plcJam>	optional	Beinhaltet die Grundinformation für das Einrichtungs-Ereignis 'Stau' . Detaillierte Beschreibung in Abschnitt 2.3.1.12.
<plcOperatorRequiredStarted>	optional	Beinhaltet die Grundinformation für das Einrichtungs-Ereignis 'Bediener angefordert (Start)' . Detaillierte Beschreibung in Abschnitt 2.3.1.13.
<plcOperatorRequired>	optional	Beinhaltet die Grundinformation für das Einrichtungs-Ereignis 'Bediener angefordert' . Detaillierte Beschreibung in Abschnitt 2.3.1.14.

<plcShiftChanged>	optional	Beinhaltet die Grundinformation für das Einrichtungs-Ereignis 'Schicht gewechselt' . Detaillierte Beschreibung in Abschnitt 2.3.1.15.
<plcChargeChanged>	optional	Veraltet, nur noch wegen Kompatibilitätsgründen enthalten. Beinhaltet die Grundinformation für das Einrichtungs-Ereignis 'Charge gewechselt' . Detaillierte Beschreibung in Abschnitt 2.3.1.16.
<plcMaterialChangeStarted>	optional	Beinhaltet die Grundinformation für das Einrichtungs-Ereignis 'Beginn Rüsten Material' . Detaillierte Beschreibung in Abschnitt 2.3.1.17.
<plcMaterialChanged>	optional	Beinhaltet die Grundinformation für das Einrichtungs-Ereignis 'Material Gerüstet' . Detaillierte Beschreibung in Abschnitt 2.3.1.18.
<plcToolChangedStarted>	optional	Beinhaltet die Grundinformation für das Einrichtungs-Ereignis 'Beginn Rüsten Werkzeug' . Detaillierte Beschreibung in Abschnitt 2.3.1.19.
<plcToolChanged>	optional	Beinhaltet die Grundinformation für das Einrichtungs-Ereignis 'Werkzeug Gerüstet' . Detaillierte Beschreibung in Abschnitt 2.3.1.20.
<plcLogIn>	optional	Beinhaltet die Grundinformation für das Einrichtungs-Ereignis 'Benutzer hat sich angemeldet' . Detaillierte Beschreibung im Abschnitt 2.3.1.21.
<plcLogOff>	optional	Beinhaltet die Grundinformation für das Einrichtungs-Ereignis 'Benutzer hat sich abgemeldet' . Detaillierte Beschreibung im Abschnitt 2.3.1.22.
<partReceived>	optional	Beinhaltet die Grundinformationen für das Prozess-Ereignis 'Werkstück angekommen' . Detaillierte Beschreibung in Abschnitt 2.3.2.1.
<partStateChanged>	optional	Veraltet. Wird in der aktuellen Spezifikation nicht mehr unterstützt.
<partProcessingStarted>	optional	Beinhaltet die Grundinformationen für das Prozess-Ereignis 'Bearbeitung gestartet' . Detaillierte Beschreibung in Abschnitt 2.3.2.3.
<partProcessingPaused>	optional	Beinhaltet die Grundinformationen für das Prozess-Ereignis 'Bearbeitung unterbrochen' . Detaillierte Beschreibung in Abschnitt 2.3.2.4.
<partProcessingAborted>	optional	Beinhaltet die Grundinformationen für das Prozess-Ereignis 'Bearbeitung abgebrochen' . Detaillierte Beschreibung in Abschnitt 2.3.2.5.
<partProcessed>	optional	Beinhaltet die Grundinformationen für das Prozess-Ereignis 'Bearbeitung abgeschlossen' . Detaillierte Beschreibung in Abschnitt 2.3.2.6.
<partDisplaced>	optional	Beinhaltet die Grundinformationen für das Prozess-Ereignis 'Werkstück umgesetzt' . Detaillierte Beschreibung in Abschnitt 2.3.2.7.
<dataDownloadRequired>	optional	Beinhaltet die Grundinformationen für das Daten-Ereignis 'Daten fehlen' . Detaillierte Beschreibung in Abschnitt 2.3.3.1.

<dataUploadRequired>	optional	Beinhaltet die Grundinformationen für das Daten-Ereignis ' Neue Daten vorhanden '. Detaillierte Beschreibung in Abschnitt 2.3.3.2.
<result>	optional	Beinhaltet das Ergebnis der Bearbeitung des Ereignisses durch MES. Dieses Element kann nur in der Antwort von MES enthalten sein.
<trace>	optional	Beinhaltet detaillierte Meldungen bzw. alle Fehler / Warnungen die während der Verarbeitung aufgetreten sind. Dieses Element kann nur in der Antwort von MES enthalten sein. Detaillierte Beschreibung in Abschnitt 2.2.1.6.

2.2.1.5 <result>-Element

Beinhaltet das Ergebnis der Bearbeitung des Ereignisses durch MES. Wird nur in den Antwort-Nachrichten von MES verwendet.

Attribute

Attribut	Angabe	Datentyp	Kommentar
returnCode	zwingend	DINT	Rückgabewert von MES. Ist der Inhalt dieses Attributs '0', so wurde das Ereignis erfolgreich bearbeitet.

Unterelemente

-

Inhalt

Beschreibung des Ergebnisses (optional).



Beispiel

```
<result returnCode="0" />
```

```
<result returnCode="75443">
  Unable to transmit data to FMS.
</result>
```

2.2.1.6 <trace>-Element

Neben dem oben beschriebenen Mechanismus der Fehlerrückmeldung, kann über die Verwendung des <trace>-Elements auch mehrere Fehler von MES zurückgemeldet werden. Das <trace>-Element kann nur in der Antwort enthalten sein und liefert die Verarbeitungsergebnisse der einzelnen Verarbeitungsschritte eines Ereignisses zurück. Das <trace>-Array ist innerhalb des <event>-Elements und kann die Rückgabewerte aller Verarbeitungsschritte enthalten.



Hinweis

Da SPS-Clients keine dynamischen Speicheroperationen ausführen können, kann die max. Anzahl von Einträgen in diesem Feld limitiert sein. Die Grenzwerte können der Dokumentation der Client-Bibliotheken entnommen werden.

Attribute

Attribut	Angabe	Datentyp	Kommentar
level	zwingend	STRING	Trace-Level des Eintrags. Zurzeit möglich: <ul style="list-style-type: none"> ○ Error ○ Warning
code	zwingend	DINT	Rückgabewert des Verarbeitungsschritts
text	zwingend	STRING	Kann Zusatzinformation zum Rückgabewert enthalten. Ansonsten wird ein Leerstring gesendet.
source	zwingend	STRING	Quelle des Rückgabewerts. Beschreibt typischerweise den MES-Dienst.

Wird das `<trace>`-Element benutzt, kann der Rückgabewert der Nachricht (`result@returnCode`) die folgenden Werte enthalten:

- **-2**
Bei der Verarbeitung des Ereignisses sind Warnmeldungen aufgetreten, jedoch **keine** Fehler.
- **-1**
Bei der Verarbeitung des Ereignisses sind Fehler aufgetreten. Die einzelnen Fehler können im `<trace>`-Feld gefunden werden.
- **0**
Die Verarbeitung des Ereignisses war erfolgreich. Es sind weder Fehler noch Warnmeldungen aufgetreten.
- **> 0**
Rückgabewerte > 0 behalten ihre bisherige Bedeutung bei (Kompatibilitäts-Modus). Bei diesen Rückgabewerten kann keine Aussage über den Inhalt des `<trace>`-Feldes gemacht werden. Bei einer solchen Antwort, ist die Verarbeitung nicht vollständig erfolgreich gewesen, es kann jedoch nicht zwischen Fehler und Warnmeldungen unterschieden werden.

**Hinweis**

Diese Option steht nur für OpCon Xml V2.0 zur Verfügung. Clients die diese Version nicht unterstützen, können das Trace-Feld nicht auswerten. Ebenso können serverseitig Dienste bzw. Verarbeitungsmodule existieren, die dieses Feature noch nicht unterstützen.

**Beispiel**

Die folgenden Beispiele beziehen sich auf ein Ereignis, welches serverseitig drei Verarbeitungsschritte (vMDT und MatControl) auslöst.

MES meldet einen Fehler zurück:

```
<event>
  <result returnCode="-1">MES detected an error.</result>
  <trace>
    <trace level="error" code="18666" text="Some text" source="vMDT"/>
  </trace>
</event>
```

MES meldet eine Warnung zurück:

```
<event>
  <result returnCode="-2">MES detected warning.</result>
  <trace>
    <trace level="warning" code="32001" text="Some text" source="MatControl"/>
  </trace>
</event>
```

MES meldet erfolgreiche Verarbeitung:

```
<event>
  <result returnCode="0"></result>
  <trace/>
</event>
```

MES meldet Fehler und Warnung:

```
<event>
  <result returnCode="-1">MES detected an error.</result>
  <trace>
    <trace level="error" code="18666" text="Some text" source="vMDT"/>
    <trace level="warning" code="32001" text="Some text" source="MatControl"/>
  </trace>
</event>
```

2.2.1.7 <body>-Element

Dieses Element beinhaltet die applikationsspezifischen Daten. Die Unterelemente dieses Elements sind frei wählbar. In Teil II der Spezifikation werden Elemente definiert, welche innerhalb des <body>-Elements für die Übertragung spezifischer Informationen benutzt werden können. Die Elemente im <body> dienen zur Anpassung des Ereignisses auf die jeweilige datentechnische Verarbeitung des Prozesses. Die Daten innerhalb des <body>-Elements können dann z. B. für `partProcessed` bei Prozess A in die OpCon Ergebnisdatenbank übertragen werden. Für Prozess B muss jedoch bei `partProcessed` zusätzlich noch Daten in eine dat-Datei geschrieben werden. Damit können diese zusätzlichen Daten von Prozess B nur im <body>-Element für Prozess B aufgeführt werden um die Schnittstelle von Prozess A nicht unnötig aufzublähen.

2.3 Abbildung der Ereignisse

2.3.1 Abbildung der Einrichtungs-Ereignisse

2.3.1.1 <plcChangeOverStarted>-Element

Das Ereignis `plcChangeOverStarted` wird im Allgemeinen von Einrichtungen vor bzw. mit dem Beginn des Umrüstvorganges ausgelöst. Es beinhaltet die allgemeingültigen Informationen zu diesem Einrichtungsereignis.

Attribute

Attribut	Angabe	Datentyp	Kommentar
typeNo	zwingend	STRING(10)	10-stellige Typnummer des Typs auf den umgerüstet werden soll.
typeVar	optional	STRING(10)	10-stellige Typvariante auf die umgerüstet werden soll. Existiert keine Typvariante, so kann dieses Attribut entweder weggelassen werden, oder <code>typeVar</code> muss einen Leerstring ("") enthalten.
typeVersion	optional	STRING	Version des Typs auf den umgerüstet werden soll.

			Existiert keine Typversion, so kann dieses Attribut entweder weggelassen werden, oder <code>typeVersion</code> muss einen Leerstring ("") enthalten.
--	--	--	---

**Beispiel**

```
<plcChangeOverStarted typeNo="1111111111"
                        typeVar="0815"
                        typeVersion="V3.23"
/>
```

2.3.1.2 <plcChangeOver>-Element

Das Ereignis `plcChangeOver` wird im Allgemeinen von Einrichtungen nach dem Beenden des Umrüstvorgangs ausgelöst. Es beinhaltet die allgemeingültigen Informationen zu diesem Ereignis.

Attribute

Attribut	Angabe	Datentyp	Kommentar
typeNo	zwingend	STRING(10)	10-stellige Typnummer des aktuell gefahrenen Typs.
typeVar	optional	STRING(10)	10-stellige Typvariante des aktuell gefahrenen Typs. Existiert keine Typvariante, so kann dieses Attribut entweder weggelassen werden, oder <code>typeVar</code> muss einen Leerstring ("") enthalten.
typeVersion	optional	STRING	Version des aktuellen Typs. Existiert keine Typversion, so kann dieses Attribut entweder weggelassen werden, oder <code>typeVersion</code> muss einen Leerstring ("") enthalten.

**Beispiel**

```
<plcChangeOver typeNo="1111111111"
                typeVar=""
                typeVersion="V3.23"
/>
```

2.3.1.3 <plcOperationModeChanged>-Element

Das Ereignis `plcOperationModeChanged` wird im Allgemeinen von Einrichtungen nach dem Wechseln der Betriebsart ausgelöst. Es beinhaltet die allgemeingültigen Informationen zu diesem Ereignis.

Attribute

Attribut	Angabe	Datentyp	Kommentar
typeNo	optional	STRING(10)	10-stellige Typnummer des aktuell gefahrenen Typs. Existiert keine Typnummer, so kann dieses Attribut entweder weggelassen werden, oder <code>typeNo</code> muss einen Leerstring ("") enthalten.
typeVar	optional	STRING(10)	10-stellige Typvariante des aktuell gefahrenen Typs. Existiert keine Typvariante, so kann dieses Attribut entweder weggelassen werden, oder <code>typeVar</code> muss einen Leerstring ("") enthalten.

typeVersion	optional	STRING	Version des aktuellen Typs. Existiert keine Typversion, so kann dieses Attribut entweder weggelassen werden, oder typeVersion muss einen Leerstring ("") enthalten.
shift	optional	DINT	Aktuelle Schichtnummer. Existiert keine Schichtnummer, so kann dieses Attribut entweder weggelassen werden, oder shift muss 0 enthalten.
charge	optional	STRING	Kennung der aktuell gefertigten Charge. Soll oder kann die Charge nicht verwendet werden, so muss hier ein Leerstring ("") angegeben werden.
specPrgNo	optional	DINT	Nummer des aktuell durchgeführten Sonderbetrieb-Programms. Wenn kein solches Programm läuft muss der Inhalt dieses Attributs '0' sein.
operationMode	zwingend	INT	Nummer der aktuellen Betriebsart. Die Kodierung der Betriebsart muss anhand der bestehenden Definitionen des OpCon-Standards vorgenommen werden.
modeOn	zwingend	BOOL	Betriebsart aktiv Dieses Attribut beinhaltet die Information ob die Betriebsart aktiv ist oder nicht: <ul style="list-style-type: none"> • true - Betriebsart ist aktiv • false - Betriebsart nicht aktiv

**Beispiel**

```

<plcOperationModeChanged typeNo="111111111"
    typeVar=""
    shift="1"
    charge=""
    specPrgNo="0"
    operationMode="1"
    modeOn="true"
/>

```

2.3.1.4 <plcEventOn>-Element

Veraltet.

Wird in der aktuellen Version nicht mehr unterstützt.

2.3.1.5 <plcEventOff>-Element

Veraltet.

Wird in der aktuellen Version nicht mehr unterstützt.

2.3.1.6 <plcSystemStarted>-Element

Das Ereignis `plcSystemStarted` wird im Allgemeinen von Einrichtungen unmittelbar nach dem Hochfahren des Systems ausgelöst. Es beinhaltet die allgemeingültigen Informationen zu diesem Ereignis.

Attribute

Attribut	Angabe	Datentyp	Kommentar
typeNo	optional	STRING(10)	10-stellige Typnummer des aktuell gefahrenen Typs. Existiert keine Typnummer, so kann dieses Attribut entweder weggelassen werden, oder <code>typeNo</code> muss einen Leerstring ("") enthalten.
typeVar	optional	STRING(10)	10-stellige Typvariante des aktuell gefahrenen Typs. Existiert keine Typvariante, so kann dieses Attribut entweder weggelassen werden, oder <code>typeVar</code> muss einen Leerstring ("") enthalten.
typeVersion	optional	STRING	Version des aktuellen Typs. Existiert keine Typversion, so kann dieses Attribut entweder weggelassen werden, oder <code>typeVersion</code> muss einen Leerstring ("") enthalten.



Beispiel

```
<plcSystemStarted typeNo="1111111111"
                  typeVar=""
                  typeVersion="V3.23"
/>
```

2.3.1.7 <plcStationSwitchedOff>-Element

Das Ereignis `plcStationSwitchedOff` wird im Allgemeinen von Einrichtungen unmittelbar vor dem Herunterfahren des Systems ausgelöst. Es beinhaltet keine weiteren Informationen.

Attribute

-



Beispiel

```
<plcStationSwitchedOff />
```

2.3.1.8 <plcError>-Element

Das Ereignis `plcError` wird im Allgemeinen von Einrichtungen unmittelbar nach dem Auftreten bzw. Quittieren eines Fehlers ausgelöst. Es beinhaltet die allgemeingültigen Informationen zu diesem Einrichtungsereignis.

Attribute

Attribut	Angabe	Datentyp	Kommentar
typeNo	optional	STRING(10)	10-stellige Typnummer des aktuell gefahrenen Typs. Existiert keine Typnummer, so kann dieses Attribut entweder weggelassen werden, oder <code>typeNo</code> muss einen Leerstring ("") enthalten.
typeVar	optional	STRING(10)	10-stellige Typvariante des aktuell gefahrenen Typs. Existiert keine Typvariante, so kann dieses Attribut entweder weggelassen werden, oder <code>typeVar</code> muss einen Leerstring ("") enthalten.

typeVersion	optional	STRING	Version des aktuellen Typs. Existiert keine Typversion, so kann dieses Attribut entweder weggelassen werden, oder typeVersion muss einen Leerstring ("") enthalten.
errorNo	zwingend	DINT	Fehlernummer 1..??? Fehlernummer 0 entspricht der Quittierung des Fehlers.
errorText	zwingend	STRING	Beschreibungstext für den Fehler. Steht kein Text zur Verfügung, so muss ein Leerstring ("") angegeben werden.
errorType	zwingend	INT	Klassifizierung des Fehlers: <ul style="list-style-type: none"> • 1 -> Fehler • 2 -> Warnung • 3 -> Information
errorState	optional	INT	Status des Fehlers <ul style="list-style-type: none"> • 0 -> Fehler steht an • 1 -> Fehler quittiert
operationMode	optional	INT	Nummer der aktuellen Betriebsart. Die Kodierung der Betriebsart muss anhand der bestehenden Definitionen des OpCon-Standards vorgenommen werden.
modeOn	optional	BOOL	Betriebsart aktiv Dieses Element beinhaltet die Information ob die Betriebsart aktiv ist oder nicht: <ul style="list-style-type: none"> • true - Betriebsart ist aktiv • false - Betriebsart nicht aktiv
chainNo	optional	DINT	Schrittkettennummer Dieses Element muss die Nummer der Schrittkeite enthalten, in welcher der Fehler aufgetreten ist.

**Beispiel**

```

<plcError typeNo="111111111"
  typeVar=""
  errorNo="40312"
  errorText="Ventil klemmt"
  errorType="1"
  errorState="0"
  operationMode="1"
  modeOn="true"
  chainNo="12"
/>

```

2.3.1.9 <plcPartsMissingStarted>-Element

Das Ereignis `plcPartsMissingStarted` wird im Allgemeinen von Einrichtungen unmittelbar nach dem Auftreten eines Teilemangels ausgelöst. Es beinhaltet die allgemeingültigen Informationen zu diesem Einrichtungsereignis.

Attribute

Attribut	Angabe	Datentyp	Kommentar
missingParts	zwingend	UDINT	Bit-codierte Information, welche Teile für die Durchführung des Prozesses fehlen. Das LSB entspricht dem Hauptteil, alle anderen Bits können applikationsspezifisch für Zubauteile definiert werden. Für die fehlenden Teile müssen die jeweiligen Bits auf '1' gesetzt werden.
typeNo	optional	STRING(10)	10-stellige Typnummer des aktuell gefahrenen Typs. Existiert keine Typnummer, so kann dieses Attribut entweder weggelassen werden, oder <code>typeNo</code> muss einen Leerstring ("") enthalten.
typeVar	optional	STRING(10)	10-stellige Typvariante des aktuell gefahrenen Typs. Existiert keine Typvariante, so kann dieses Attribut entweder weggelassen werden, oder <code>typeVar</code> muss einen Leerstring ("") enthalten.
typeVersion	optional	STRING	Version des aktuellen Typs. Existiert keine Typversion, so kann dieses Attribut entweder weggelassen werden, oder <code>typeVersion</code> muss einen Leerstring ("") enthalten.



Beispiel

```
<plcPartsMissingStarted missingParts="1" />
```

2.3.1.10 <plcPartsMissing>-Element

Das Ereignis `plcPartsMissing` wird im Allgemeinen von Einrichtungen unmittelbar nach dem Beseitigen eines Teilemangels ausgelöst. Es beinhaltet die allgemeingültigen Informationen zu diesem Ereignis.

Attribute

Attribut	Angabe	Datentyp	Kommentar
missingParts	zwingend	UDINT	Bit-codierte Information, welche Teile für die Durchführung des Prozesses fehlen. Das LSB entspricht dem Hauptteil, alle anderen Bits können applikationsspezifisch für Zubauteile definiert werden. Für die fehlenden Teile müssen die jeweiligen Bits auf '1' gesetzt werden.
typeNo	optional	STRING(10)	10-stellige Typnummer des aktuell gefahrenen Typs. Existiert keine Typnummer, so kann dieses Attribut entweder weggelassen werden, oder <code>typeNo</code> muss einen Leerstring ("") enthalten.
typeVar	optional	STRING(10)	10-stellige Typvariante des aktuell gefahrenen Typs. Existiert keine Typvariante, so kann dieses Attribut entweder weggelassen werden, oder <code>typeVar</code> muss einen Leerstring ("") enthalten.
typeVersion	optional	STRING	Version des aktuellen Typs. Existiert keine Typversion, so kann dieses Attribut entweder weggelassen werden, oder <code>typeVersion</code> muss einen Leerstring ("") enthalten.

**Beispiel**

```
<plcPartsMissing missingParts="1" />
```

2.3.1.11 <plcJamStarted>-Element

Das Ereignis `plcJamStarted` wird im Allgemeinen von Einrichtungen unmittelbar nach dem Auftreten eines Staus innerhalb der Einrichtung ausgelöst. Es beinhaltet die allgemeingültigen Informationen zu diesem Einrichtungsereignis.

Attribute

Attribut	Angabe	Datentyp	Kommentar
typeNo	optional	STRING(10)	10-stellige Typnummer des aktuell gefahrenen Typs. Existiert keine Typnummer, so kann dieses Attribut entweder weggelassen werden, oder <code>typeNo</code> muss einen Leerstring ("") enthalten.
typeVar	optional	STRING(10)	10-stellige Typvariante des aktuell gefahrenen Typs. Existiert keine Typvariante, so kann dieses Attribut entweder weggelassen werden, oder <code>typeVar</code> muss einen Leerstring ("") enthalten.
typeVersion	optional	STRING	Version des aktuellen Typs. Existiert keine Typversion, so kann dieses Attribut entweder weggelassen werden, oder <code>typeVersion</code> muss einen Leerstring ("") enthalten.

**Beispiel**

```
<plcJamStarted typeNo="1111111111" />
```

2.3.1.12 <plcJam>-Element

Das Ereignis `plcJam` wird im Allgemeinen von Einrichtungen unmittelbar nach dem Beseitigen eines Staus innerhalb der Einrichtung ausgelöst. Es beinhaltet die allgemeingültigen Informationen zu diesem Einrichtungsereignis.

Attribute

Attribut	Angabe	Datentyp	Kommentar
typeNo	optional	STRING(10)	10-stellige Typnummer des aktuell gefahrenen Typs. Existiert keine Typnummer, so kann dieses Attribut entweder weggelassen werden, oder <code>typeNo</code> muss einen Leerstring ("") enthalten.
typeVar	optional	STRING(10)	10-stellige Typvariante des aktuell gefahrenen Typs. Existiert keine Typvariante, so kann dieses Attribut entweder weggelassen werden, oder <code>typeVar</code> muss einen Leerstring ("") enthalten.
typeVersion	optional	STRING	Version des aktuellen Typs. Existiert keine Typversion, so kann dieses Attribut entweder weggelassen werden, oder <code>typeVersion</code> muss einen Leerstring ("") enthalten.

**Beispiel**

```
<plcJam typeNo="1111111111" />
```

2.3.1.13 <plcOperatorRequiredStarted>-Element

Das Ereignis `plcOperatorRequiredStarted` wird im Allgemeinen von Einrichtungen unmittelbar nach dem Auftreten eines Bedienerrufs ausgelöst. Es beinhaltet die allgemeingültigen Informationen zu diesem Einrichtungsereignis.

Attribute

Attribut	Angabe	Datentyp	Kommentar
operator	zwingend	UDINT	Bit-codierte Information, welche(r) Bediener angefordert wurde(n) (Anlagenführer,...).
typeNo	optional	STRING(10)	10-stellige Typnummer des aktuell gefahrenen Typs. Existiert keine Typnummer, so kann dieses Attribut entweder weggelassen werden, oder <code>typeNo</code> muss einen Leerstring ("") enthalten.
typeVar	optional	STRING(10)	10-stellige Typvariante des aktuell gefahrenen Typs. Existiert keine Typvariante, so kann dieses Attribut entweder weggelassen werden, oder <code>typeVar</code> muss einen Leerstring ("") enthalten.
typeVersion	optional	STRING	Version des aktuellen Typs. Existiert keine Typversion, so kann dieses Attribut entweder weggelassen werden, oder <code>typeVersion</code> muss einen Leerstring ("") enthalten.

**Beispiel**

```
<plcOperatorRequiredStarted operator="1" />
```

2.3.1.14 <plcOperatorRequired>-Element

Das Ereignis `plcOperatorRequired` wird im Allgemeinen von Einrichtungen unmittelbar nach dem Eintreffen der angeforderten Person ausgelöst. Es beinhaltet die allgemeingültigen Informationen zu diesem Einrichtungsereignis.

Attribute

Attribut	Angabe	Datentyp	Kommentar
operator	zwingend	UDINT	Bit-codierte Information, welche(r) Bediener jetzt bei der Einrichtung angekommen ist.
typeNo	optional	STRING(10)	10-stellige Typnummer des aktuell gefahrenen Typs. Existiert keine Typnummer, so kann dieses Attribut entweder weggelassen werden, oder <code>typeNo</code> muss einen Leerstring ("") enthalten.
typeVar	optional	STRING(10)	10-stellige Typvariante des aktuell gefahrenen Typs. Existiert keine Typvariante, so kann dieses Attribut entweder weggelassen werden, oder <code>typeVar</code> muss einen Leerstring ("") enthalten.

typeVersion	optional	STRING	Version des aktuellen Typs. Existiert keine Typversion, so kann dieses Attribut entweder weggelassen werden, oder <code>typeVersion</code> muss einen Leerstring ("") enthalten.
-------------	----------	--------	---

**Beispiel**

```
<plcOperatorRequired operator="1" />
```

2.3.1.15 <plcShiftChanged>-Element

Das Ereignis `plcShiftChanged` wird im Allgemeinen von Einrichtungen unmittelbar nach dem Schichtwechsel ausgelöst. Es beinhaltet die allgemeingültigen Informationen zu diesem Ereignis.

Attribute

Attribut	Angabe	Datentyp	Kommentar
shiftNo	zwingend	DINT	Aktuelle Schichtnummer nach dem Schichtwechsel.

**Beispiel**

```
<plcShiftChanged shiftNo="1" />
```

2.3.1.16 <plcChargeChanged>-Element

Das Ereignis `plcChargeChanged` wird im Allgemeinen von Einrichtungen unmittelbar nach einem Chargenwechsel ausgelöst. Es beinhaltet die allgemeingültigen Informationen zu diesem Ereignis.

Attribute

Attribut	Angabe	Datentyp	Kommentar
charge	zwingend	STRING	Aktuelle Chargennummer nach dem Wechsel.

**Beispiel**

```
<plcChargeChanged shiftNo="GFKM889AA56" />
```

2.3.1.17 <plcMaterialChangeStarted>-Element

Dieses Ereignis wird vom Fertigungsprozess gesendet, unmittelbar bevor ein Materialwechsel stattfindet. Der Fertigungsprozess hat mit diesem Ereignis die Möglichkeit vorab Information über das neue Material abzufragen, um damit ein unnötiges Umrüsten zu vermeiden (z. B. wenn neues Material gesperrt ist). Auf MES-Seite wird diese Ereignis typischerweise so verarbeitet, dass die Materialverwaltung über das neue Material abgefragt wird und diese Informationen an den Fertigungsprozess gesendet werden.

Attribute

Attribut	Angabe	Datentyp	Kommentar
identifier	zwingend	STRING	ID des neuen Materials
oldIdentifier	optional	STRING	ID des alten (=aktuell gerüsteten) Materials
oldQuantity	optional	REAL	Restmenge des alten (=aktuell gerüsteten) Materials.
name	optional	STRING	Eindeutiger Name des Materials wie bereits beim Rüsten übermittelt.



Beispiel

```
<plcMaterialChangeStarted identifier="884566S123456789012"
oldIdentifier="884566S123456786001"
oldQuantity="17.0"
name="PowerConnector"
/>
```

2.3.1.18 <plcMaterialChanged>-Element

Dieses Ereignis wird vom Fertigungsprozess gesendet, unmittelbar nachdem ein Materialwechsel stattgefunden hat. Der Fertigungsprozess bestätigt über dieses Ereignis, dass das neue Material aufgerüstet wurde.

Auf MES-Seite wird dieses Ereignis typischerweise so verarbeitet, dass das neue Material in der Material-Verwaltung dem Fertigungsprozess zugeordnet wird.

Attribute

Attribut	Angabe	Datentyp	Kommentar
identifier	zwingend	STRING	ID des neuen (=aktuell gerüsteten) Materials
oldIdentifier	optional	STRING	ID des alten Materials
oldQuantity	optional	REAL	Restmenge des alten Materials.
name	optional	STRING	Eindeutiger Name des Materials wie bereits beim Rüsten übermittelt.



Beispiel

```
<plcMaterialChanged identifier="884566S123456789012"
oldIdentifier="884566S123456786001"
oldQuantity="17.0"
name="PowerConnector"
/>
```

2.3.1.19 <plcToolChangeStarted>-Element

Dieses Ereignis wird vom Fertigungsprozess gesendet, unmittelbar bevor ein Werkzeugwechsel stattfindet. Der Fertigungsprozess hat mit diesem Ereignis die Möglichkeit vorab Information über das neue Werkzeug abzufragen, um damit ein unnötiger Umbau zu vermeiden (z. B. wenn neues Werkzeug gesperrt ist).

Auf MES-Seite wird diese Ereignis typischerweise so verarbeitet, dass die Werkzeugverwaltung über das neue Werkzeug abgefragt wird und diese Informationen an den Fertigungsprozess gesendet werden.

Attribute

Attribut	Angabe	Datentyp	Kommentar
identifier	zwingend	STRING	ID des neuen Werkzeugs
oldIdentifier	optional	STRING	ID des alten (=aktuell gerüsteten) Werkzeugs
name	optional	STRING	Eindeutiger Name des Werkzeugs wie bereits beim Rüsten übermittelt.

**Beispiel**

```
<plcToolChangeStarted identifier="932478970022"
                        oldIdentifier="932478970001"
                        name="Mask"
/>
```

2.3.1.20 <plcToolChanged>-Element

Dieses Ereignis wird vom Fertigungsprozess gesendet, unmittelbar nachdem ein Werkzeugwechsel stattgefunden hat. Der Fertigungsprozess bestätigt über dieses Ereignis, dass das neue Werkzeug aufgerüstet wurde.

Auf MES-Seite wird dieses Ereignis typischerweise so verarbeitet, dass das neue Werkzeug in der Werkzeug-Verwaltung dem Fertigungsprozess zugeordnet wird.

Attribute

Attribut	Angabe	Datentyp	Kommentar
identifier	zwingend	STRING	ID des neuen (= aktuell gerüsteten) Werkzeugs
oldIdentifier	optional	STRING	ID des alten Werkzeugs
name	optional	STRING	Eindeutiger Name des Werkzeugs wie bereits beim Rüsten übermittelt.

**Beispiel**

```
<plcToolChanged identifier="932478970022"
                 oldIdentifier="932478970001"
                 name="Mask"
/>
```

2.3.1.21 <plcLogIn>-Element

Dieses Ereignis wird vom Fertigungsprozess gesendet, wenn ein Benutzer sich explizit anmeldet.

Typischerweise wird dieses Ereignis auf MES-Seite so verarbeitet, dass der Benutzer über den AuthenticationService authentifiziert wird.

In der Antwort kann dann MES dem Fertigungsprozess mitteilen, ob die Authentifizierung erfolgreich war. Falls auch der Fertigungsprozess ein Berechtigungskonzept besitzt, kann MES in der Antwort zusätzlich noch die Rollen übertragen, die dem Benutzer zugeordnet sind.

Attribute

Attribut	Angabe	Datentyp	Kommentar
user	zwingend	STRING	Kennung (z. B. LogIn) des Benutzers
pwd	zwingend	STRING	Passwort (verschlüsselt) des Benutzers. Der Verschlüsselungsalgorithmus muss dem verschlüsselten Passwort vorangestellt werden. Getrennt werden die beiden Informationen durch einen ':' [Algorithmus][:][Verschlüsseltes Passwort]

**Beispiel**

```
<plcLogIn user="xyz2zz"
          pwd="md5:a934x75yy23"
/>
```

2.3.1.22 <plcLogOff>-Element

Dieses Ereignis wird vom Fertigungsprozess gesendet, wenn ein Benutzer sich explizit abmeldet. Typischerweise wird dieses Ereignis dann auf MES-Seite so verarbeitet, dass sämtliche Anfrage die von dieser Location kommen dann wieder im Kontext des Default-Benutzers laufen.

Attribute

-

**Beispiel**

```
<plcLogOff />
```

2.3.2 Abbildung der Prozess-Ereignisse**2.3.2.1 <partReceived>-Element**

Das Ereignis `partReceived` wird im Allgemeinen von Einrichtungen unmittelbar nach dem Ankommen eines neuen Werkstücks ausgelöst. Es beinhaltet die allgemeingültigen Informationen zu diesem Prozess-Ereignis.

Attribute

Attribut	Angabe	Datentyp	Kommentar
identifizier	zwingend	STRING	Identifikation des Werkstücks
typeNo	optional	STRING(10)	Typnummer auf welche die Station / der Prozess aktuell gerüstet ist.
typeVar	optional	STRING(10)	Typvariante auf welche die Station / der Prozess aktuell gerüstet ist.
typeVersion	optional	STRING	Typversion auf welche die Station / der Prozess aktuell gerüstet ist.

**Beispiel**

```
<partReceived identifier="PC-20050420-1300-445" typeNo="1111111111" typeVar=""
              typeVersion=""
/>
```

2.3.2.2 <partStateChanged>-Element

Veraltet.

Wird in der aktuellen Version nicht mehr unterstützt.

2.3.2.3 <partProcessingStarted>-Element

Das Ereignis `partProcessingStarted` wird im Allgemeinen von Einrichtungen unmittelbar nach dem Beginn der Bearbeitung eines Werkstücks ausgelöst. Es beinhaltet die allgemeingültigen Informationen zu diesem Prozess-Ereignis.

Attribute

Attribut	Angabe	Datentyp	Kommentar
identifizier	zwingend	STRING	Identifikation des Werkstücks

**Beispiel**

```
<partProcessingStarted identifizier="PC-20050420-1300-445" />
```

2.3.2.4 <partProcessingPaused>-Element

Das Ereignis `partProcessingPaused` wird im Allgemeinen von Einrichtungen ausgelöst, wenn die Bearbeitung eines Werkstücks unterbrochen wurde. Es beinhaltet die allgemeingültigen Informationen zu diesem Prozess-Ereignis.

Attribute

Attribut	Angabe	Datentyp	Kommentar
identifizier	zwingend	STRING	Identifikation des Werkstücks

**Beispiel**

```
<partProcessingPaused identifizier="PC-20050420-1300-445" />
```

2.3.2.5 <partProcessingAborted>-Element

Das Ereignis `partProcessingAborted` wird im Allgemeinen von Einrichtungen unmittelbar nach dem (vorzeitigen) Abbruch der Bearbeitung eines Werkstücks ausgelöst. Es beinhaltet die allgemeingültigen Informationen zu diesem Prozess-Ereignis.

Attribute

Attribut	Angabe	Datentyp	Kommentar
identifizier	zwingend	STRING	Identifikation des Werkstücks

**Beispiel**

```
<partProcessingAborted identifizier="PC-20050420-1300-445" />
```

2.3.2.6 <partProcessed>-Element

Das Ereignis `partProcessed` wird im Allgemeinen von Einrichtungen unmittelbar nach der Bearbeitung eines Werkstücks ausgelöst. Es beinhaltet die allgemeingültigen Informationen zu diesem Prozess-Ereignis.

Attribute

Attribut	Angabe	Datentyp	Kommentar
identifizier	zwingend	STRING	Identifikation des Werkstücks



Beispiel

```
<partProcessed identifizier="PC-20050420-1300-445" />
```

2.3.2.7 <partDisplaced>-Element

Das Ereignis `partDisplaced` wird im Allgemeinen von Einrichtungen unmittelbar nach dem Umsetzen eines Werkstücks ausgelöst. Es beinhaltet die allgemeingültigen Informationen zu diesem Prozess-Ereignis.

Attribute

Attribut	Angabe	Datentyp	Kommentar
oldIdentifizier	zwingend	STRING	Bisherige Identifikation des Werkstücks
identifizier	zwingend	STRING	Neue Identifikation des Werkstücks



Beispiel

```
<partDisplaced oldIdentifizier="PC-20050420-1300-445"
  identifizier="IX-20050420-1300-445" />
```

2.3.3 Abbildung der Daten-Ereignisse

2.3.3.1 <dataDownloadRequired>-Element

Das Ereignis `dataDownloadRequired` wird im Allgemeinen von Einrichtungen immer dann ausgelöst, wenn besondere Daten von Server angefordert werden müssen. Es beinhaltet keine weiteren Informationen.

Attribute

-



Beispiel

```
<dataDownloadRequired />
```

2.3.3.2 <dataUploadRequired>-Element

Das Ereignis `dataUploadRequired` wird im Allgemeinen von Einrichtungen immer dann ausgelöst, wenn besondere Daten an den Server übermittelt werden müssen. Es beinhaltet keine weiteren Informationen.

Attribute

-



Beispiel

```
<dataUploadRequired />
```

Kommunikationsablauf

Tritt ein Ereignis auf der Station auf, so wird dies über eine XML-Nachricht an MES gesendet. Diese Nachricht kann sowohl die allgemeinen Ereignisinformationen als auch die applikationsspezifischen Daten (z. B. Messergebnisse) beinhalten. Der Empfang dieser Nachricht wird über das verwendete Netzwerkprotokoll abgesichert. Eine explizite Empfangsnachricht des MES zur Station ist nicht vorgesehen. Der MES kann nun gemäß seiner Konfiguration beginnen, dass Ereignis zu bearbeiten. Nach erfolgreicher Bearbeitung wird der Station das Gesamtergebnis der Bearbeitung ebenfalls über TCP/IP übertragen. Als Antwort verwendet MES das von der Station gesendete Telegramm und spiegelt die Kopfinformationen. Zusätzlich wird dem `<event>`-Elemente noch das Element `<result>` hinzugefügt. Dieses Element enthält das Ergebnis der Verarbeitung des Ereignisses durch MES.

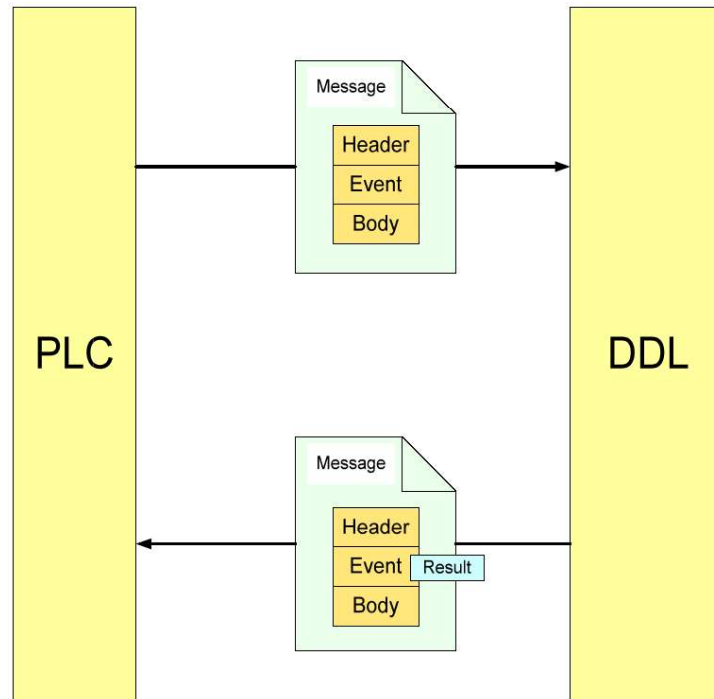


Abbildung 3 Genereller Ablauf der Kommunikation

Müssen Daten auf die Station übertragen werden, so können diese Daten ebenfalls an das Telegramm im `<body>`-Element der Antwort angehängt werden.



Beispiel

Station 10 wird vom Handbetrieb in den Automatikbetrieb geschaltet. Dabei wird von der Station die folgende Nachricht an MES geschickt um dort das Betriebsart-Ereignis mitzuteilen:

```

<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="1" version="2.0" eventName="plcOperationModeChanged"
    timeStamp="2005-04-03T13:21:34.231+02:00">
    <location lineNo="1" statNo="10" statIdx="1" processName="FLASH"
      application="PLC" />
  </header>
  <event>
    <plcOperationModeChanged operationMode="1" modeOn="true" />
  </event>
</root>
  
```

Dieses Ereignis wird empfangen und verarbeitet. Die Station erhält dann als Quittierung der Verarbeitung die folgende Nachricht:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="1" version="2.0" eventName="plcOperationModeChanged"
    timeStamp="2005-04-03T13:21:34.231+02:00">
    <location lineNo="1" statNo="10" statIdx="1" processName="FLASH"
      application="PLC" />
    </header>
  <event>
    <result returnCode="0" />
  </event>
</root>
```

Der im Beispiel gezeigte Ablauf gilt für jedes Ereignis. Sollen in der Antwort zusätzliche Daten an die Station übertragen werden, so enthält die Antwort noch das `<body>`-Element. Dieses `<body>`-Element enthält dann die zu übertragenden Daten.

3 Teil 2: Erweiterte Elemente

Die hier beschriebenen Elemente sind für spezifische Anforderungen an den Informationsaustausch zwischen Server und Clients definiert. Auf Clientseite können diese Elemente je nach Leistungsfähigkeit des Clients implementiert werden oder nicht. Implementiert ein Client ein Element, so muss dieses Element jedoch vollständig implementiert werden. Eine teilweise Implementierung ist nicht zulässig. Auf Serverseite müssen diese Elemente implementiert werden. Dies betrifft jedoch nur die Kommunikationsschnittstelle. Es ist zulässig, dass aufgrund der nachgelagerten Verarbeitung verschiedene optionale Elemente nicht für den Datenaustausch zur Verfügung stehen. Daher ist bei der Verwendung dieser Elemente stets die Dokumentation der Server und der Client-Seite zu überprüfen, ob das Element auch zur Verfügung steht.



Hinweis

Die hier definierten Elemente müssen nicht von allen Clients implementiert werden. Es kommt auf die Leistungsfähigkeit und Nutzen an, ob die Elemente implementiert werden oder nicht.

3.1 <items>-Element

Das <items>-Element ist der Container für applikationsspezifische Datenelemente (Variablen). Es kann nur als unmittelbares Unterelement des <body>-Elements verwendet werden. Innerhalb des <body>-Elements darf nur ein <items>-Element existieren.

Attribute

-

Unterelemente

Element	Angabe	Kommentar
<item>	zwingend	Repräsentiert ein einzelner Wert der Daten (z. B. ein Rüstparameter)



Beispiel

Das folgende Beispiel enthält ein <items>-Element welches die Rüstparameter für eine Station enthält.

```
<items>
  <item name="Param1" value="3.0E-99" dataType="5" />
  <item name="Param2" value="3.22088E+9" dataType="5" />
  <item name="Param3" value="-5.998776E+1" dataType="5" />
  <item name="Param4" value="Hello" dataType="8" />
  <item name="Param5" value="1000" dataType="3" />
  <item name="Param6" value="true" dataType="11" />
</items>
```

3.1.1 <item>-Element

Dieses Element repräsentiert ein einzelner applikationsspezifischer Wert (Rüstparameter, Messwert,...). Es kann nur innerhalb des <items>-Element verwendet werden. Dort kann es beliebig oft angegeben werden.

Attribute

Attribut	Angabe	Datentyp	Kommentar
name	zwingend	STRING	Name des Werts (z. B. Name der Variable des Rüstparameters innerhalb der SPS)
value	zwingend	variabel	Inhalt des Werts
dataType	zwingend	DINT	Datentyp des Werts (z. B. Datentyp des Rüstparameters DINT,

			STRING, real,...). Der Datentyp wird als OpCon Datentyp codiert (siehe Datamanagement.chm -> VarType).
--	--	--	---

**Beispiel**

```
<item name="ResVar1" value="3.299E-1" dataType="5" />
```

3.2 <resHead>-Element

Das <resHead>-Element enthält die wichtigsten Grunddaten über das bearbeitete Werkstück. Die darin enthaltenen Informationen geben die Kurzübersicht über das Ergebnis des Prozesses und den Grunddaten des Werkstückes. Dieses Element kann nur innerhalb des <body>-Elements verwendet werden und darf dort nur ein Mal vorkommen.

Attribute

Attribut	Angabe	Datentyp	Kommentar
result	zwingend	INT	Gesamtergebnis des Prozesses bezogen auf das Werkstück.
typeNo	zwingend	STRING(10)	Typnummer des Werkstücks
typeVar	optional	STRING(10)	Typvariante des Werkstücks
typeVersion	optional	STRING	Typversion des Werkstücks
charge	optional	STRING	Charge des Werkstücks (veraltet, in neuen Projekten nicht mehr verwenden)
workingCode	optional	INT	Bearbeitungscode (Serienteil, Prüfteil, Masterteil,...)
nioBits	zwingend	DINT	Bitcodierte Information beim Auftreten eines Fehlers. Diese können zur Steuerung der einzuleitenden Maßnahme auf einem Nacharbeitsplatz verwendet werden.
workCycleCount	optional	DINT	Bearbeitungszähler (Nacharbeitszähler)
cycleTimePrev	optional	DINT	Taktzeit vorangegangenes Teil in Millisekunden.
orderID	optional	STRING	Fertigungs-Auftragsnummer
batch	optional	STRING	Charge des Werkstücks (Ersetzt resHead.charge)
machineID	optional	STRING	Eindeutige ID der Maschine. Diese ID ist der Maschine gekoppelt und nicht an den Standort der Maschine wie z. B. Stationsnummer usw.

**Beispiel**

```
<resHead
  result="1"
  typeNo="0281011900"
  typeVar="0009"
  batch=""
  workingCode="4"
  nioBits="0"
  workCycleCount="1"
  cycleTimePrev="9"
  orderID="ABB342"
  machineID="C709"
/>
```


3.3 Benutzerdefinierte Elemente

Innerhalb des <body>-Elements können benutzerdefinierte Elemente definiert werden. Über den Aufbau und Inhalt dieser Elemente wird keine Annahme gemacht. Es muss jedoch sichergestellt werden, dass die Clients die solche Telegramme senden bzw. empfangen sollen, die Bedeutung der Elemente kennen und entsprechenden interpretieren können (z. B. Datentyp der Attribute).

3.3.1 <structs>-Element

Das <structs>-Element kapselt alle Datenstruktur-Elemente.



Hinweis

Dieses Element kann nur verwendet werden, wenn `header@contentType` den Wert 2 oder 3 enthält.

Attribute

-

Unterelemente

-



Beispiel

Soll der Ergebnisdaten-Kopf übertragen werden, so kann dies entweder wie oben beschrieben direkt innerhalb des <body>-Elements erfolgen, oder die Struktur kann innerhalb des <structs>-Elements durch Setzen des `contentType` auf 3 (oder 2) übertragen werden:

`contentType = 3`

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="1" version="2.0" eventName="partProcessed" contentType="3">
    <location lineNo="1" statNo="10" statIdx="1" processName="FLASH"
      application="PLC" />
  </header>
  <event>
    <partProcessed identifier="xyx" />
  </event>
  <body>
    <structs>
      <resHead result="1" typeNo="0281011900" workingCode="4" nioBits="0" />
    </structs>
  </body>
</root>
```

`contentType = 1 (oder nicht vorhanden)`

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="1" version="2.0" eventName="partProcessed" contentType="1">
    <location lineNo="1" statNo="10" statIdx="1" processName="FLASH"
      application="PLC" />
  </header>
  <event>
    <partProcessed identifier="xyx" />
  </event>
  <body>
    <resHead result="1" typeNo="0281011900" workingCode="4" nioBits="0" />
  </body>
</root>
```

3.4 Felder

3.4.1 <arrays>-Element (Standard-Felder)

Das <arrays>-Element enthält die Felder (Arrays) die innerhalb der XML-Nachricht übertragen werden sollen. Es können nur Felder übertragen werden, deren Elemente einfache Datentypen (z. B. DINT, STRING, ...) besitzen. Geschachtelte Felder werden nicht unterstützt.

Alle Felder müssen innerhalb des <arrays>-Element definiert werden. Dieses <arrays>-Element darf innerhalb des <body>-Elements nur einmal existieren. Pro Standard-Feld enthält <arrays> ein entsprechendes <array>-Element. Der Name und Datentyp des Standard-Felds wird als Attribute `name` und `dataType` angegeben.



Beispiel

```
<arrays>
  <array name="sample" dataType="5">
    <item value="0.0" />
    <item value="1.1" />
    <item value="2.2" />
    <item value="3.3" />
    <item value="4.4" />
  </array>
  <array name="test" dataType="3">
    <item value="0" />
    <item value="11" />
    <item value="22" />
    <item value="33" />
    <item value="44" />
  </array>
</arrays>
```

Die einzelnen Elemente eines Felds werden durch <item>-Elemente dargestellt.

3.4.2 <structArrays>-Element (Struktur Felder)

Das <structArrays>-Element enthält die Felder deren Elemente komplexe Datentypen besitzen (Arrays von Strukturen). Es können mehrere solcher Felder pro XML-Nachricht versendet werden.



Hinweis

Da eine SPS keine dynamischen Speicheroperationen unterstützt, ist die maximale Anzahl der übertragbaren Felder pro Nachricht begrenzt. Die aktuellen Grenzwerte sind in den jeweiligen Dokumentationen der SPS-Bausteine hinterlegt.



Hinweis

Dieses Element kann nur verwendet werden, wenn `header@contentType` den Wert 2 oder 3 enthält.

Attribute

-

Unterelemente

Element	Angabe	Kommentar
<array>	zwingend	Repräsentiert ein zu übertragendes Feld

<array>-Element

Dieses Element enthält das zu übertragende Feld. Es kann nur innerhalb des <structArrays>-Element verwendet werden. Dort kann es beliebig oft angegeben werden.

Attribute

Attribut	Angabe	Datentyp	Kommentar
name	zwingend	STRING	Name des Felds. Der Name des Felds muss eindeutig sein (es dürfen keine zwei <arrays>-Elemente mit gleichem Namen in einer XML-Nachricht existieren).

Unterelemente

Element	Angabe	Kommentar
<structDef>	zwingend	Enthält den Aufbau des komplexen Datentyps der als Feld übertragen werden soll.
<values>	zwingend	Enthält die Nutzdaten des Felds.

<structDef>-Element

Dieses Element enthält die Definition des Aufbaus des komplexen Datentyps der als Feld übertragen werden soll. Jedes Element der Datenstruktur wird dabei durch ein <item>-Element repräsentiert.

Attribute

-

Unterelemente

Element	Angabe	Kommentar
<item>	zwingend	Beschreibt ein Element der Datenstruktur.

<item>-Element

Das <item>-Element innerhalb <structDef> beschreibt ein einzelnes Element der Datenstruktur. Es können mehrere dieser <item>-Elemente innerhalb <structDef> enthalten sein.

Attribute

Attribut	Angabe	Datentyp	Kommentar
name	zwingend	STRING	Name des Elements Dieser Name wird unter <values> verwendet um das Element zu referenzieren.
dataType	zwingend	DINT	Datentyp des Elements

<values>-Element

Dieses Element enthält die eigentlichen Nutzdaten des Felds.

Unterelemente

Element	Angabe	Kommentar
<item>	zwingend	Enthält den Wert eines Array-Elements

Attribute

-

<item>-Element

Das <item>-Element innerhalb <values> beschreibt die einzelnen Elemente des Felds (Array). Es können mehrere dieser <item>-Elemente innerhalb <values> enthalten sein.

Attribute

Die Attribute werden durch die Einträge innerhalb `<structDef>` festgelegt. Jedes `<item>` innerhalb `<structDef>` wird als Attribut eines `<item>`-Eintrags innerhalb von `<values>` interpretiert.

Die Struktur-Arrays haben damit folgenden Aufbau:

```
<structArrays>
  <array name="">
    <structDef>
      <item name="[valA]" dataType="" />
      <item name="[valB]" dataType="" />
      (...)
    </structDef>
    <values>
      <item valA="" valB="" />
      <item valA="" valB="" />
      (...)
    </values>
  </array>

  <array name="">
    <structDef>
      <item name="[valC]" dataType="" />
      <item name="[valD]" dataType="" />
      (...)
    </structDef>
    <values>
      <item valC="" valD="" />
      <item valC="" valD="" />
      (...)
    </values>
  </array>
  (...)
</structArrays>
```

**Beispiel**

```
<structArrays>
  <array name="Tools">
    <structDef>
      <item name="id" dataType="8" />
      <item name="state" dataType="3" />
      <item name="counter" dataType="3" />
    </structDef>
    <values>
      <item id="0003243_001" state="0" counter="119999" />
      <item id="2343245_345" state="0" counter="65434" />
    </values>
  </array>
  <array name="Component">
    <structDef>
      <item name="matId" dataType="8" />
      <item name="state" dataType="3" />
    </structDef>
    <values>
      <item matId="884566S123456786001" state="0" />
      <item matId="884566S123456789012" state="0" />
    </values>
  </array>
</structArrays>
```

3.4.3 Benutzerdefinierte Felder

Innerhalb des `<body>`-Elements können benutzerdefinierte Elemente definiert werden. Ein solches Feld besteht aus einem Container-Element und den Feld-Elementen. Um das Container-Element als Container für ein Feld zu kennzeichnen muss das Element mit dem Attribut `isArray` ausgezeichnet werden. Dem Attribut muss der Wert `true` zugewiesen werden. Innerhalb des Containers liegen dann die Elemente des Felds. Die Namen der Feld-Elemente müssen identisch mit dem Namen des Container-Elements sein. Die Feld-Elemente dürfen keine weiteren Unter-Elemente enthalten. Sie dürfen jedoch eine beliebige Anzahl von Attributen besitzen, die sich jedoch nicht von Feld-Element zu Feld-Element unterscheiden dürfen.



Beispiel

In diesem Beispiel wird ein Feld `userDef` definiert. Jedes Element des Felds soll die Information `identifier` und `internalName` beinhalten.

```
<root>
  <body>
    <userDef isArray="true">
      <userDef identifier="xyz" internalName="name1" />
      <userDef identifier="abc" internalName="name2" />
    </userDef>
  </body>
</root>
```

Anhang A: Vorgabewerte

Result

Für die Bewertung eines Prozessergebnisses bzw. für die Bewertung der Prozessdurchführung ansich, stehen die folgenden Werte zur Verfügung.

Wert	Beschreibung
-1	Kein Status
0	Nicht gemessen
1	Gut
2	Schlecht
3	Abbruch
4	Ergebniswert zu klein
5	Ergebniswert zu groß
6	Range zu groß
7	Zeitüberschreitung
8	Stringvergleich ist falsch
9	Gemessen
10	Ausreisser
11	Information wird nicht verwendet
12	Verschrottet
255	Unvollständig eingetragen

CheckType

Die Behandlung und Eingabe von Parameterwerten hängt im Wesentlichen vom Toleranztyp des jeweiligen Parameters ab. Dieser Toleranztyp wird durch den Wert von CheckType definiert. Es sind die folgenden Werte zulässig:

Wert	Symbol	Bewertungskriterium	Beschreibung
1	<=	Ergebnis <= oberem Grenzwert	Das Ergebnis wird mit IO bewertet, wenn der Ergebniswert kleiner oder gleich dem in UpLim angegebenen Wert ist. Dies setzt voraus, dass in UpLim der obere Grenzwert als Absolutwert angegeben wurde.
2	>=	Ergebnis >= unterem Grenzwert	Das Ergebnis wird mit IO bewertet, wenn der Ergebniswert grösser oder gleich dem in LowLim angegebenen Wert ist. Dies setzt voraus, dass in LowLim der untere Grenzwert als Absolutwert angegeben wurde.
3	SetValue oder SetStr	Sollwert	Eine Bewertung des Ergebnisses findet nicht statt. Es handelt sich um einen reinen Vorgabewert.
4	res=SetValue±LowLim	Ergebnis innerhalb relativer Toleranz	Das Ergebnis wird mit IO bewertet, wenn der Ergebniswert innerhalb der spezifizierten symmetrischen (relativen) Toleranz (also z.B. +/- 0.5 bar) liegt. Dies setzt voraus, dass in LowLim der absolut Wert der relativen Toleranz (also z.B. 0.5 bar) angegeben wurde. Es ist nicht möglich asymmetrische relative Toleranzen (also z.B. -0.2 bar als Untergrenze und +0.5 bar als Obergrenze) anzugeben.

5	LowLim<=res<=UpLim	Ergebnis innerhalb absoluter Toleranz	Das Ergebnis wird mit IO bewertet, wenn der Ergebniswert innerhalb der in LowLim bzw. UpLim spezifizierten absoluten Toleranzen (z.B. untere Grenze 12.5 bar und obere Grenze 13.2 bar) liegt. Dies setzt voraus, dass sowohl in LowLim als auch in UpLim die absolute Toleranz angegeben ist.
6	res<=LowLim,res>=UpLim	Ergebnis ausserhalb des Bereichs	Das Ergebnis wird mit IO bewertet, wenn der Ergebniswert ausserhalb des durch LowLim bzw. UpLim definierten Bereichs liegt. Dabei werden LowLim bzw. UpLim noch als gültige Werte anerkannt. Dies setzt voraus, dass sowohl in LowLim als auch in UpLim der auszuschliessende Bereich in absoluter Form angegeben wurde (also z.B. LowLim = 12.3 bar UpLim = 13.5 bar)
7	res=SetStr	Ergebnis gleich Vorgabe (case sensitive)	Das Ergebnis wird mit IO bewertet, wenn der Ergebniswert gleich dem Vorgabewert ist. ACHTUNG: Nur für STRING gültig. Es wird Gross-/Kleinschreibung unterschieden (case sensitive).
8	res/RES=SetStr	Ergebnis gleich Vorgabe (case insensitive)	Das Ergebnis wird mit IO bewertet, wenn der Ergebniswert gleich dem Vorgabewert ist. ACHTUNG: Nur für STRING gültig. Es wird Gross-/Kleinschreibung nicht unterschieden (case insensitive). Landesspezifische Sonderzeichen werden weiterhin in Gross-/Kleinschreibung unterschieden (ä ist nicht gleich Ä)
9	res=SetValue-LowLim	Ergebnis innerhalb negativer relativer Toleranz	Das Ergebnis wird mit IO bewertet, wenn der Ergebniswert zwischen Sollwert unterer Grenze liegt. Die untere Grenze wird dabei relativ in LowLim angegeben.
10	res=SetValue+UpLim	Ergebnis innerhalb positiver relativer Toleranz	Das Ergebnis wird mit IO bewertet, wenn der Ergebniswert zwischen Sollwert oberer Grenze liegt. Die obere Grenze wird dabei relativ in UpLim angegeben.
11	res=SetValue-LowLim%	Ergebnis innerhalb negativer relativer Toleranz	Das Ergebnis wird mit IO bewertet, wenn der Ergebniswert zwischen Sollwert unterer Grenze liegt. Die untere Grenze wird dabei prozentual in LowLim angegeben.
12	res=SetValue+UpLim%	Ergebnis innerhalb positiver relativer Toleranz	Das Ergebnis wird mit IO bewertet, wenn der Ergebniswert zwischen Sollwert oberer Grenze liegt. Die obere Grenze wird dabei prozentual in UpLim angegeben.
13	res=SetValue±LowLim%	Ergebnis innerhalb relativer Toleranz	Das Ergebnis wird mit IO bewertet, wenn der Ergebniswert innerhalb der spezifizierten symmetrischen (relativen) Toleranz liegt. Dies setzt voraus, dass in LowLim der prozentuale Wert der Toleranz angegeben wurde. Es ist nicht möglich asymmetrische Toleranzen anzugeben.

Werte für OperationMode

Für das melden von Betriebszuständen sind die folgenden Vorgabewerte definiert:

Wert	Beschreibung
1	Automatikbetrieb
2	Schrittbetrieb
3	Handbetrieb
4	Sonderbetrieb

WorkingCode

Der Bearbeitungscode drückt aus, die Bearbeitungsbedingungen (z. B. Serienbetrieb, Kalibrierung, usw.) aus. Die folgenden Werte sind definiert:

Wert	Beschreibung
0	Serienteil
1	Testteil
2	Musterteil
3	Reparaturteil
4	Kalibrierteil
5	Masterteil
6	Stabilitätsteil
7	Umrüstteil
8	Datenaustausch (zwischen Stationen, Programmen,...)
9	Leer-WT
10	Teil für CG-Messung
11	Teil für SM-Messung
12	Audit-WT
13	Teil für GRR-Messung
14	Aufwärm-WT
15	Golden Device

PartState

Diese Vorgabewerte drücken den Zustand des Werkstückes aus. Folgende Werte sind definiert:

Wert	Beschreibung
0	Kein Teil
1	Nicht für Station
2	Neues Teil
3	Teil in Bearbeitung
4	Teil IO
5	Teil NIO
6	Teil schreiben
7	Ergebnis geschrieben
8	Nacharbeitsteil
9	Teil nicht vorhanden
10	Teil für Bearbeitung
11	Teil vorhanden