OCIT
Open Communication Interface for Road Traffic Control Systems
Offene Schnittstellen für die Straßenverkehrstechnik

OCIT-Outstations
Basisfunktionen für Feldgeräte

OCIT-O-Basis_V1.1_A02

OCIT® Developer Group (ODG)
OCIT® ist eine registrierte Marke der Firmen Dambach, Siemens, Signalbau Huber, STOYE und Stührenberg
OCIT-Outstations

Basisfunktionen für Feldgeräte
Inhaltsverzeichnis

1 Einführung ................................................................................................................... 7
2 Spezielle Definitionen ....................................................................................................... 7
   2.1 Systemzeit ................................................................................................................. 7
   2.2 Erkennen von Störungen des Übertragungsweges ..................................................... 7
      2.2.1 Unterscheiden von Übertragungsstörungen ................................................. 8
   2.3 Vorgangskennung, SYSJOBID ............................................................................... 9
   2.4 Zähleweise bei nummerierten Elementen ............................................................... 12
3 Systemzugänge ................................................................................................................ 13
   3.1 Zentraler Systemzugang ........................................................................................... 13
   3.2 Lokaler Systemzugang ............................................................................................. 14
4 Objektdefinitionen ........................................................................................................... 14
   4.1 Systemobjekte .......................................................................................................... 14
      4.1.1 Systemobjekt Feldgerät .......................................................................... 15
      4.1.2 Systemobjekt Zentrale ................................................................................ 17
      4.1.3 Systemobjekt Remote Device ................................................................. 17
      4.1.4 RemoteService ........................................................................................... 19
   4.2 Meldungen und Messwerte (Archive). ...................................................................... 20
      4.2.1 Eigenschaften der Archive .......................................................................... 20
      4.2.2 Die Archivschnittstelle .............................................................................. 21
      4.2.3 Elemente der Archivschnittstelle ............................................................... 23
      4.2.4 Liste ............................................................................................................ 24
      4.2.5 Sekundenframe / Auftragsframe ................................................................. 36
      4.2.6 Auftrag ....................................................................................................... 37
      4.2.7 Meldung ..................................................................................................... 43
4.2.8 Meldungsteil ................................................................. 44
4.2.9 Welche Archive existieren? ............................................. 47
4.2.10 Verhalten bei Stromausfall ........................................... 48
4.2.11 Übertragungsformat von Archivdaten (Format der Meldung) ........................................ 48
4.2.12 Elementbeschreibungen Meldungsarchiv ..................... 48

5 Abläufe Meldung und Messwerte ........................................... 51
  5.1 Listen mit vordefinierten Aufträgen ........................................ 51
    5.1.1 Ziel ........................................................................ 51
    5.1.2 Ablauf .................................................................... 51
  5.2 Veränderung von Listen .................................................. 52
  5.3 Wechseln des Grades (der Wichtigkeit) einzelner Meldungen .......... 52
  5.4 Abholen von Daten ........................................................... 53
    5.4.1 Kontinuierliches Abholen von Daten ................................ 53
    5.4.2 Spontanes Abholen von Teilen des Ringpuffers .................. 54
  5.5 Feststellen, ob die Liste von außen (Systemzugang etc.) oder während einer Systemstörung geändert wurde .................................. 54
  5.6 Ändern eines Auftrags während des Laufes ......................... 55
  5.7 Getriggertes Abholen von Daten ......................................... 55
  5.8 Abholen eines Datensatzes sofort nach Auftreten .................. 56
  5.9 Synchronisieren nach Übertragungsstörung ...................... 56
### Dokumentenstand

<table>
<thead>
<tr>
<th>Version Zustand</th>
<th>Verteilerkreis</th>
<th>Datum</th>
<th>Kommentar</th>
</tr>
</thead>
<tbody>
<tr>
<td>V 1.0</td>
<td>PUBLIC</td>
<td>6. September 2002</td>
<td>1. freigegebene Ausgabe</td>
</tr>
<tr>
<td>V1.1 A01</td>
<td>PUBLIC</td>
<td>15. Juli 2004</td>
<td>Text an verschiedenen Stellen aktualisiert und korrigiert. Kapitel überarbeitet: 2.1 Systemzeit, 2.2.1.2 Methode OnNetzAus (schnelle Meldung bei Netzausfall), 3 Systemzugänge, 4.1.1 Systemobjekt Feldgerät, 4.2.4.3 Eventhandler für Listen in der Zentrale, 4.2.6.2 Meldungsauftrag, 4.2.12 Erweiterung der Türmeldungen, 4.2.4.3 Eventhandler (Methode OnNetzAus) Kapitel neu: 2.4 Zählweise bei nummerierten Elementen</td>
</tr>
<tr>
<td>V1.1 A02</td>
<td>PUBLIC</td>
<td>17. 11. 2004</td>
<td>Event OnNetzAus() ergänzt mit Vorgangskennung (Kapitel 2.2.1.2 und 4.2.4.3, Methode 203).</td>
</tr>
</tbody>
</table>

### Referenzdokumente OCIT-Outstations, Stand 17. 11. 2004

<table>
<thead>
<tr>
<th>Gültig</th>
<th>Dokumente</th>
<th>Titel</th>
<th>Datenspezifikationen (XML-Dateien)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generell</td>
<td>OCIT-O-System_V1.1</td>
<td>Einführung in das System</td>
<td></td>
</tr>
<tr>
<td></td>
<td>OCIT-O-Protokoll_V1.1</td>
<td>Regeln und Protokolle</td>
<td>OCIT-O-DTD_V1.0.dtd</td>
</tr>
<tr>
<td></td>
<td>OCIT-O-Basis_V1.1_A02</td>
<td>Basisdefinitionen für Feldgeräte ¹</td>
<td>OCIT-O-Basis-TYPE_V1.1_A02.xml</td>
</tr>
<tr>
<td>Speziell</td>
<td>OCIT-O-Lstg_V1.1_A02</td>
<td>Lichtsignalsteuergeräte</td>
<td>OCIT-O-Lstg-TYPE_V1.1.xml</td>
</tr>
<tr>
<td>Optional</td>
<td>OCIT-O- Profil_1_V1.1</td>
<td>Profil 1 – Übertragungsprofil für</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Punkt-zu-Punkt-Verbindungen auf</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>festgeschalteten Übertragungswe-</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>gen</td>
<td></td>
</tr>
</tbody>
</table>

¹ Geräte, deren Einsatzort die Straße ist, wie Lichtsignalsteuerungen, Verkehrsmessstellen oder Anzeigesteuerungen, werden in der OCIT-Standardisierung generalisierend als Feldgeräte bezeichnet.
### Abkürzungen

<table>
<thead>
<tr>
<th>Abkürzung</th>
<th>Definition / Standard in</th>
</tr>
</thead>
<tbody>
<tr>
<td>bps</td>
<td>bits per second (= bit/s)</td>
</tr>
<tr>
<td>BTPPL</td>
<td>Basis Transport Paket Protokoll Layer</td>
</tr>
<tr>
<td>HDLC</td>
<td>High level Data Link Protocol</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>OCIT</td>
<td>Open Communication Interface for Road Traffic Control Systems</td>
</tr>
<tr>
<td>OSI</td>
<td>Open Systems Interconnection</td>
</tr>
<tr>
<td>PPP</td>
<td>Point to Point Protocol</td>
</tr>
<tr>
<td>RFC</td>
<td>Request for Comment (= Arbeitspapiere, Protokoll-Spezifikationen oder Kommentare zu Netzwerk-Themen)</td>
</tr>
<tr>
<td>SHA-1</td>
<td>Secure Hash Algorithm</td>
</tr>
<tr>
<td>TCP</td>
<td>Transmission Control Protocol</td>
</tr>
<tr>
<td>UDP</td>
<td>User Datagram Protocol - low end transport service</td>
</tr>
<tr>
<td>V.xx</td>
<td>Standards der ITU-T (International Telecommunications Union), früher CCITT</td>
</tr>
<tr>
<td>XML</td>
<td>eXtensible Markup Language, Herstellerunabhängige Auszeichnungssprache, mit der u.a. eine Schnittstellenbeschreibung verteilster Applikationen realisiert werden kann (spezifiziert durch W3C)</td>
</tr>
</tbody>
</table>
1 Einführung

In diesem Dokument finden sich Definitionen zu Funktionen, die in Lichtsignalsteuerungen aber auch in Verkehrsmessstellen und anderen typischen Feldgeräten der Straßenverkehrstechnik in ähnlicher Art und Weise vorhanden sind, wie zum Beispiel Archive oder die Mel- 
dungen „Tür auf“, „Störung“ etc.

Es ist nicht verpflichtend, dass OCIT-Outstations konforme Geräte alle hier festgelegten Funktionen unterstützen. Sie unterstützen nur diejenigen Funktionen, die für den jeweiligen Zweck und Ausbau notwendig sind.

Die Definitionen gelten für Feldgeräte und Zentralen.

2 Spezielle Definitionen

2.1 Systemzeit

Die Lichtsignalsteuergeräte besitzen lokale Uhren. Ihre genaue Einstellung ist Sache der Feldgeräte, die dazu den Zeitdienst NTP (RFC 1305) der Zentrale nutzen können. Der Zeit- 
dienst gleicht durch die Übertragungszeit zwischen Zentrale und Feldgerät bedingte Zeitfehler aus. Weitere Festlegungen siehe Dokument OCIT-O Protokoll.

Zusätzlich ist eine direkte Abfrage der Gerätezeit durch die Zentrale und Abfrage der Zentra- lenzeit durch das Feldgerät möglich (siehe Systemobjekt Feldgerät 4.1.1 und Systemobjekt Zentrale 4.1.2). Diese Abfragen sind mit einem durch die Übertragungszeit zwischen Zentrale und Feldgerät bedingten Zeitfehler behaftet.

2.2 Erkennen von Störungen des Übertragungsweges

(Ergänzung durch Definitionen in den Dokumenten OCIT-O Profil_nn möglich)

Unter „Störung des Übertragungsweges“ wird hier ein vollständiger Ausfall der Übertra- 
gungsstrecke über mehrere Sekunden, wie er bei einer Unterbrechung der Verbindung oder einem Ausfall der Versorgungsspannung („Netzausfall“) auftreten kann, verstanden. Übertra- 
gungsstörungen werden auf defekte Systemteile zurückgeführt. Sporadische Übertragungsstö- 
rungen können temporär zu vergleichbaren Störungsbildern führen.

Übertragungsstörungen können verursacht werden durch:

- Ausfall des zentralen Rechners oder des Feldgerätes
- Ausfall von Übertragungseinrichtungen in Feldgerät oder Zentrale
- Untерbrechung des Übertragungsweges
- Netzausfall
Derartige Übertragungsstörungen werden durch das Fehlen von Telegrammen erkannt, wobei die Schnelligkeit der Erkennung abhängig ist von der Häufigkeit der abgeschickten Telegramme:

- **Erkennungsmöglichkeit in der Zentrale:**
  Auf die Telegramme der Zentrale erfolgt keine Antwort.

- **Erkennungsmöglichkeit im Feldgerät:**
  Es treffen keine Telegramme der Zentrale ein.

In den OCIT-Outstations Definitionen sind für Schaltbefehle Start- und Endezeiten vorgesehen, die das Feldgerät bei fehlenden Telegrammen veranlassen, in Vorzugsprogramme zu schalten. Darüber hinaus können in regelmäßigen Abständen Kontrolltelegramme gesendet werden, bei deren ausbleiben auf eine Übertragungsstörung geschlossen werden kann.

Erkennungsmöglichkeiten, die auf Funktionen des Übertragungsgerätes beruhen, wie z.B. Trägerüberwachung, sind hier nicht aufgeführt, da sie typabhängig sind und die Übertragungsgeräte in OCIT-Outstations nicht vorgeschrieben werden.

### 2.2.1 Unterscheiden von Übertragungsstörungen

Vom Feldgerät erkannte Übertragungsstörungen erzeugen die OCIT-Outstations Meldung *Kommunikationsstörung* (4.2.12) und gelangen über die Abfrage des Standard-Meldearchivs (4.2.9) zur Zentrale.


#### 2.2.1.1 Unterscheidung nach Beseitigen der Störung

Nach Beseitigen der Kommunikationsstörung werden bei ursächlichem Netzausfall OCIT-Outstations Meldungen aus dem Feldgerät zur Zentrale übertragen (Standard-Meldearchiv), die der Zentrale eine Eingrenzung der Ursache im Nachhinein erlauben:

- **Netz aus** mit Angabe des Ausfallzeitpunkts (4.2.12)
- **Netz ein** (4.2.12)
- **Kommunikation ok** (4.2.12)

Die Zentrale kann damit im Nachhinein den ursächlichen Netzausfall erkennen und die ursprüngliche Meldung „Kommunikationsstörung“ eingrenzen.

#### 2.2.1.2 Unterscheidung sofort nach Auftreten

Diese Option erlaubt den Feldgeräten die sofortige Meldung eines Netzausfalls und damit der Zentrale eine sofortige Eingrenzung der Meldung „Kommunikationsstörung“ auf den vorliegenden Fehlerfall.
Dazu benötigen die Feldgeräte eine Pufferung der Versorgungsspannung (Kurzzeit-USV), um die für die Meldung benötigten Geräteteile über die notwendige Zeit der Melderoutine weiter mit Spannung zu versorgen. Es werden zwei Methoden zur Übermittlung der Information über den Netzausfall definiert, die sich in der Länge der notwendigen Pufferzeit der Versorgungsspannung unterscheiden.

Variante a) Netzausfall-Meldung über Standard-Meldearchiv


Variante b) Netzausfall Meldung über Eventliste


Es bleibt den Herstellern überlassen, ob sie dieses optionale Ausstattungsmerkmal „Erkennen von Netzausfall sofort nach Auftreten“ anbieten oder nicht. Wird diese Option angeboten, gelten folgende Festlegungen:


- Andere Lösungsmöglichkeiten sind nicht OCIT-Outstations konform und nur herstellerspezifisch wirksam.

2.3 Vorgangskennung, SYSJOBID


Dazu ist es nötig, dass jeder Vorgangsinitiator eine eindeutige Nummer verwendet. OCIT-Outstations verwendet fest vergebene Nummernbänder. Die Vorgangskennung ist ein Be-

Auf Grund der Vorgangskennung kann

- das Feldgerät die internen Abläufe ordnen und
- die Zentrale die Bedienvorgänge verschiedener Bediener und Systeme dokumentieren.

Die Vorgangskennung muss während der maximal zu erwartenden Laufzeit eines Bedienkommandos eindeutig sein.


Die Auftragsnummer vergibt immer der durch die Herkunftskennung bestimmte Systemteil.

Welche Vorgänge die Vorgangskennung mitführen, wird zusammen mit der Definition der Schnittstellenfunktion festgelegt.

**Vorgangskennung:**

<table>
<thead>
<tr>
<th>Vorgangskennung</th>
<th>Bezeichnung</th>
</tr>
</thead>
<tbody>
<tr>
<td>Herkunft</td>
<td>Auftragsnummer</td>
</tr>
</tbody>
</table>

**Systembezeichnung**

<table>
<thead>
<tr>
<th>Teilsystem</th>
<th>Typ</th>
<th>Untertyp</th>
<th>Instanz</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 = Zentrale</td>
<td>0 = keine Detaillierung</td>
<td>0 bis 63</td>
<td>0 bis 65535</td>
</tr>
<tr>
<td>2 = Systemzugang</td>
<td>0 = keine Detaillierung</td>
<td>0 bis 63</td>
<td>0 bis 65535</td>
</tr>
<tr>
<td>3 = Feldgerät</td>
<td>0 = keine Detaillierung</td>
<td>0 bis 65535</td>
<td>0 bis 63</td>
</tr>
</tbody>
</table>

1 = ZAUT 0 = keine Detaillierung
1 -15 Nr. der Zeitautomatik
Zeitautomatik/Schaltuhr

2 = VA-Logiken 0 = keine Detaillierung
1 - 15 Nr. der Logik
Verkehrsabhängige Logiken oder Anwenderprogramme im Feldgerät/Zentrale
<table>
<thead>
<tr>
<th>Vorgangskennung</th>
<th>Herkunft</th>
<th>Auftragsnummer</th>
<th>Bezeichnung</th>
</tr>
</thead>
<tbody>
<tr>
<td>Systembezeichnung</td>
<td>Verursacher</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Teilsystem</td>
<td>Typ</td>
<td>Untertyp</td>
<td>Instanz</td>
</tr>
<tr>
<td>3 = Bedienzugänge</td>
<td>0 = keine Detaillierung</td>
<td>1 = Bediengerät integriert</td>
<td>2 = Bediengerät abgesetzt</td>
</tr>
<tr>
<td>4 = Übertragungssysteme</td>
<td>0 = keine Detaillierung</td>
<td>1-15 für genauere Angabe des Übertragungssystemteils</td>
<td></td>
</tr>
<tr>
<td>5 = Überwachungen</td>
<td>0 = keine Detaillierung</td>
<td>1-15 für genauere Angabe der Überwachungseinrichtung</td>
<td></td>
</tr>
<tr>
<td>6 - 15 frei</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Teilsystemkennungen:**

<table>
<thead>
<tr>
<th>Kenn Nr.</th>
<th>Teilsystem</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>nicht festgelegt</td>
</tr>
<tr>
<td>1</td>
<td>Zentrale</td>
</tr>
<tr>
<td>2</td>
<td>Systemzugang</td>
</tr>
<tr>
<td>3</td>
<td>Feldgerät</td>
</tr>
</tbody>
</table>
Formate:

<table>
<thead>
<tr>
<th>2³²</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Teilsystem</td>
<td>Typ</td>
<td>Untertyp</td>
<td>Instanz</td>
<td>Auftragsnummer</td>
</tr>
<tr>
<td>2 bit</td>
<td>4 bit</td>
<td>4 bit</td>
<td>Zentrale: 6 bit</td>
<td>Zentrale: 16 bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Systemzugang: 6 bit</td>
<td>Systemzugang: 16 bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Feldgerät: 16 bit</td>
<td>Feldgerät: 6 bit</td>
</tr>
</tbody>
</table>

Beispiele:

Bedienungsvorgang von zentraler Zeitautomatik (ZAUT), Zentrale 0:

| 1 | 1 | 1 | 0 | Auftragsnummer | JAUT Tagesplan |

Bedienungsvorgang über zentrale Systemkomponente, Zentrale 0:

| 1 | 3 | 1 | 0 | Auftragsnummer | Zentrale Bedienung (manuell) |

Bedienung über lokales, abgesetztes Bediengerät (z.B. Handpanel), Feldgerät 317:

| 3 | 3 | 2 | 317 | Auftragsnummer | Abgesetztes Bediengerät |

2.4 Zählweise bei nummerierten Elementen

- Die Adressierung nummerierter Elemente wie Signalgruppen und Detektoren etc. beginnt mit dem Indexwert 1. Der Index wir nicht gemappt: Index 1 adressiert Element 1 usw. Damit ist sichergestellt, dass der Indexwert mit der von den Anwendern verwendeten Nummer eines nummerierten Elements übereinstimmt.

- Die Zählung von Zeiten beginnt mit Zeit 0. Zeit 0 bezeichnet den ersten Zeittakt von seinem Beginn bis zum seinem Ende.
3 Systemzugänge

In einem OCIT-Outstations-System sind folgende Systemzugänge vorgesehen:

- **Zentraler Systemzugang**

- **Lokaler Systemzugang**

### 3.1 Zentraler Systemzugang

Der Systemzugang in der Zentrale besteht aus einer oder mehreren Schnittstellen, die eine Kommunikation mit den Feldgeräten erlauben. Es handelt sich um logisch völlig identische Schnittstellen, wie sie auch zu den Feldgeräten führen, die Anbindung erfolgt jedoch über LAN. Eine OCIT-Zentrale muss mindestens einen Systemzugang bieten:

<table>
<thead>
<tr>
<th>Schnittstellen</th>
<th>Verbindlich</th>
<th>projektspezifisch / herstellerspezifisch</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anzahl</td>
<td>1</td>
<td>&gt; 1</td>
</tr>
<tr>
<td>Übertragungsprofil</td>
<td>LAN 10 Mbps / LAN 100 Mbps / 10/100 Base T Ethernet Stecker RJ-45</td>
<td>ISDN oder andere Dienste</td>
</tr>
<tr>
<td>Protokoll</td>
<td>OCIT wie zu den Feldgeräten</td>
<td></td>
</tr>
</tbody>
</table>

Die Hersteller können jedoch zusätzlich mehrere Zugänge vorsehen und auch ISDN-Anschlüsse oder andere Dienste bieten.

Der Betreiber / Zentralenhersteller stellt pro zentralen Systemzugang folgende Informationen zur Verfügung:

- IP Adresse des anzuschließenden Systemzugangsrechners
- IP Adresse des Gatewayrechners (falls nötig)
- IP Adresse des Namensservers (DNS)
- Vom Systemzugang zu verwendende OCIT Zentralennummer, OCIT Feldgerätenummer.

Der Systemzugang ist in erster Linie für Experten gedacht, die darüber von der Zentrale oder von entfernten Orten aus z.B. die Geräteversorgung ihrer firmeneigenen Geräte durchführen
oder Gerätetechnik testen können. Ein Feldgerät führt alle Kommandos des zugeschal-
teten Servicetools aus. Fehlbedienungen können die Systemfunktion stören. Zugang zu den
Geräten erhalten daher nur autorisierte Nutzer, die ihre Berechtigung durch die Kenntnis von
Passwörtern nachweisen müssen.

Prinzipiell ist es möglich, über den Systemzugang auch Schaltwünsche abzusetzen, welche
den Ist-Zustand des Geräts beeinflussen. Dies kann im Konflikt zu den Schaltwünschen der
Zentrale stehen. Da der Systemzugang ein Expertenzugang ist und demzufolge erwartet wird,
dass die Nutzer wissen was sie tun wird hier folgendes Verhalten definiert:

Ein gültiger Schaltwunsch wird so akzeptiert, wie wenn er von der Zentrale käme und
demzufolge auch entsprechend angesteuert. In der Zentrale wird die Zustandsänderung
über den Ist-Vektor sichtbar und über die SysJobId kann der Verursacher ermittelt wer-
den. Ein erneuter Befehl der Zentrale überschreibt dann den vorhergehenden Schalt-
wunsch erneut. Es gilt das Prinzip Last come – first serve.

3.2 Lokaler Systemzugang
Für die Anwendung „Lokaler Systemzugang“ sind derzeit noch keine Festlegungen getroffen.

4 Objektdefinitionen
Datendefinitionen siehe OCIT-O-Basis-TYPE_Vy.y.xml.

4.1 Systemobjekte

<table>
<thead>
<tr>
<th>OType</th>
<th>Name</th>
<th>Pfad (ab Feldgerät)</th>
</tr>
</thead>
<tbody>
<tr>
<td>815</td>
<td>Systemobjekt Feldgerät</td>
<td>./</td>
</tr>
<tr>
<td>817</td>
<td>Systemobjekt RemoteDevice</td>
<td>ZNr(USHORT)/FNr(USHORT)</td>
</tr>
</tbody>
</table>
### 4.1.1 Systemobjekt Feldgerät

Dieses Objekt dient dazu, allgemeine Informationen über das Feldgerät zur Verfügung zu stellen. Außerdem erlaubt es dem Feldgerät, neue Kommunikationspartner bekannt zu machen bzw. deren Passwörter zu ändern.

<table>
<thead>
<tr>
<th>Method</th>
<th>Name</th>
<th>Beschreibung</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>GetGeraeteID</td>
<td>Liest den Hersteller, Version und Gerätetyp aus. Diese Parameter können sich bei Updates ändern.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Ausgabeparameter</strong></td>
</tr>
</tbody>
</table>
|            | FgType             | 1: Zentrale  
2: Systemzugang  
3: Feldgerät                                                                                                                                                                                                 |
|            | Member : MemberID  | ID des Geräteherstellers siehe OCIT-O Protokoll                                                                                                                                                                |
|            | Devicetype : string| Typ des Geräts, welches angeschlossen ist s. Geräte-TYPE-Datei, Tag OCT.DEVICETYPE.                                                                                                                          |
|            | Version : string   | OCIT-Version, s.a. Geräte-TYPE-Datei, Tag OCT.VERSION                                                                                                                                                         |
|            | SubVersion : string| Herstellerspezifische Versionskennung, z.B. Softwareversion, s.a. Geräte-TYPE-Datei, Tag OCT.SUBVERSION                                                                                                      |
|            | APVersion : string | Version der Anwenderprogramm Software                                                                                                                                                                          |
| 101        | CreateRemoteEntry  | legt einen neuen Remote-Eintrag an                                                                                                                                                                             |
|            |                    | **Eingabeparameter**                                                                                                                                                                                          |
|            | ZNr, FNr           | ZNr, FNr des fremden Gerätes                                                                                                                                                                                  |
|            | RemoteType         | 1: Zentrale  
2: Systemzugang  
3: Feldgerät                                                                                                                                                                                                 |
|            |                    | **Ausgabeparameter**                                                                                                                                                                                          |
| 102        | DropRemoteEntry    | löscht den Remote-Eintrag                                                                                                                                                                                    |
|            |                    | **Eingabeparameter**                                                                                                                                                                                          |
|            | ZNr, FNr           | ZNr, FNr des fremden Gerätes                                                                                                                                                                                  |
|            |                    | **Ausgabeparameter**                                                                                                                                                                                          |
|            | RetCode : RetCode  | OK: Entry wurde angelegt. PARAM_INVALID falls zu löschender Eintrag nicht existiert.                                                                                                                           |
| 103        | GetTime            | gibt aktuelle Zeit des Geräts zurück                                                                                                                                                                         |
|            |                    | **Ausgabeparameter**                                                                                                                                                                                          |
### Methoden von SystemobjektFeldgeraet

<table>
<thead>
<tr>
<th>Method</th>
<th>Name</th>
<th>Beschreibung</th>
</tr>
</thead>
<tbody>
<tr>
<td>zeit</td>
<td>ZEITSTEMPEL.UTC</td>
<td>aktuelle Zeit des Gerätes</td>
</tr>
<tr>
<td></td>
<td>ZEITZONE : SLONG</td>
<td>Zeitzone (Abweichung in Sekunden von GMT)</td>
</tr>
<tr>
<td></td>
<td>ZEITQUELLE : ENUM</td>
<td>Zeitquelle (unbekannt, Quarz, Zentrale, DCF, GPS)</td>
</tr>
</tbody>
</table>

#### InstanceInfo


Diese Methode kann z.B. dazu benützt werden, alle Aufträge zu einer Liste auszulesen.

**Eingabeparameter**

- key : BaseObjType?^3 key_RefLen
  - key.Member
  - key.Otype
  - Member, OType gibt den Basisdatentyp an, von welchem die Methode Instanze Referenzen liefert.
  - key.path .. Pfadparameter je nach oben angegebenem Typ. Vom Ende des Pfades können Parameter weggelassen werden.

**Ausgabeparameter**

- RetCode ID OK Funktion wurde korrekt durchgeführt (auch wenn Klasse bekannt aber keine Instanz gefunden).
  - PARAM_INVALID falls die durch key referenzierte Klasse nicht bekannt ist.
- Path[] : paths Instanzreferenzliste bestehend aus:
  - paths.Anzahl
  - paths[].RefLen
  - paths[].Member
  - paths[].Otype
  - paths[]. ... Pfadparameter je nach zur Laufzeit angegebenem Typ.

Die OCIT Versionskennung besitzt das Format Version.Subversion, also
OCIT-Instations V1.0 „1.0“ oder „1“ (Abwärtskompatibilität)
OCIT-Instations V1.1 „1.1“
OCIT-Instations V2 „2.0“

4.1.2 Systemobjekt Zentrale
Da die Zentrale aus BTPPL-Sicht auch ein Feldgerät ist, existiert das Systemobjekt Feldgerät auch in der Zentrale.

4.1.3 Systemobjekt Remote Device

| RemoteDevice |
| Pfad: ZNr, FNr |
| METHOD | Name | Beschreibung |
| 0 | Get | Ausgabeparameter |
| | IpAdresse | Eingestellte IP-Adresse dieses Remote Devices |
| | IpName | IP Hostname dieses Remote Devices |
| | FgTyp | 1: Zentrale  
| | | 2: Systemzugang  
| | | 3: Feldgerät |
| 100 | SetPassword | Feldgeräte kennen mindestens 4 Passwörter:  
| | | - Zentrale  
| | | - Ersatzzentrale  
| | | - Systemzugang  
| | | - Default  
| | Eingabeparameter |
Das Array ist folgendermaßen aufgebaut:

\[
\text{NewPassword}[0..11] = \text{neues OCIT-Passwort}[0..11] \text{ XOR Schleier}[0..11] \\
\text{NewPassword}[12..19] = \text{Schleier}[12..19]
\]

Wenn das OCIT-Passwort kleiner als 12 Zeichen ist, wird es mit binären 0en aufgefüllt. Das Passwort wird mit einem Schleier durch eine XOR-Operation unkenntlich gemacht.

Der Schleier wird mit Hilfe des SHA-1 Algorithmus gebildet. Hierzu wird mit dem alten OCIT-Passwort des Gerätes folgende Formel ausgeführt:

\[
\text{Schleier} := \text{SHA-1( altes OCIT-Passwort + "." + ZNRZieladresse + "." + FNRZieladresse + SCHLEIER + altes OCIT-Passwort + "." + ZNRZieladresse + "." + FNRZieladresse )}
\]

Der SCHLEIER ist ein hexadezimal definiertes Array mit folgendem Inhalt:

| 00h | 49 61 65 21 20 49 61 65 21 20 50 68 20 6E 67 6C |
| 10h | 75 69 20 6D 67 6C 77 20 6E 66 68 20 43 74 68 |
| 20h | 75 6C 68 75 20 52 20 6C 79 65 68 20 77 61 67 6E |
| 30h | 20 6E 61 67 6C 20 66 68 74 61 67 6E |

Beispiel: Wenn das alte OCIT-Passwort 'OCITPASSWORT' und das Feldgerät an der Zentrale 12 mit der Gerätenummer 567 hängt, ist der Schleier die SHA-1 Prüfsumme von der Zeichenfolge

| 00h | 00h: 4F 43 49 54 50 41 53 53 57 4F 52 44 2E 31 32 2E |
| 10h | 10h: 35 36 37 49 61 65 21 20 49 61 65 21 20 50 68 20 ; 567 |
| 20h | 20h: 6E 67 6C 75 69 20 6D 67 6C 77 20 6E 66 68 20 |
| 30h | 30h: 43 74 68 75 6C 68 75 20 52 20 6C 79 65 68 20 77 |
| 40h | 40h: 61 67 6E 20 6E 61 67 6C 20 66 68 74 61 67 6E 4F ; | O
| 50h | 50h: 50 43 54 50 41 53 53 57 4F 52 44 2E 31 32 2E |
| 60h | 60h: 36 37 ; 67 |

Dadurch ändert sich der Schleier je nach Feldgerät, selbst wenn das Passwort für alle Geräte gleich ist.

Das Feldgerät berechnet denselben Schleier (mit seinem alten Passwort). Wenn dieses wieder über den Text gelegt wird, kommt das neue Passwort wieder zum Vorschein. Es wird dann bei allen weiteren Operationen eingesetzt.

Ausgabeparameter

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NO_CENTRAL: Zentralenpasswörter dürfen nur durch die Zentrale geändert werden.</td>
</tr>
</tbody>
</table>
4.1.4 RemoteService

Es gibt mehrere Quellen für steuernde Befehle, wobei jede Quelle ihre eigene Priorität hat:

1. Bedienung oder Service „lokal“
2. Bedienung oder Service „remote“
3. Zentrale
4. lokale Kommandowahl, z.B. Zeitautomatik (niedrigste Priorität)

Gedanklich entspricht ein Remote-Serviceeingriff einem verlängerten Handgerät.

Im Unterschied zu lokalen Serviceeingriffen kann das Gerät bei Remote-Serviceeingriffen das Ende nicht über den Türkontakt feststellen. Deshalb sieht OCIT einen zeitlich befristeten Befehl zum Einstellen des Servicebetriebs vor.

| RemoteService |
| Pfad: ./ (nur eine Instanz pro Feldgerät) |

<table>
<thead>
<tr>
<th>METHOD</th>
<th>Name</th>
<th>Beschreibung</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Get</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ausgabeparameter</td>
</tr>
<tr>
<td></td>
<td>EndZeit :ZEITSTEMPEL_U TC</td>
<td>Zeit bis wann die Remote Service Bedienung aktiv ist. EndZeit==0 falls keine Servicebedienung aktiv ist.</td>
</tr>
<tr>
<td></td>
<td>ServiceGrund : string</td>
<td>Gibt an warum Service Bedienung angefordert wurde.</td>
</tr>
<tr>
<td>100</td>
<td>StartService</td>
<td>Damit kann ein Service PC die Gerätebedienung anfordern. Gerät darf zu einer Zeit die Bedienberechtigung nur einmal erteilen. Falls die Remote Service Bedienung schon aktiv ist, liefert die Methode ACCESS_DENIED zurück.</td>
</tr>
<tr>
<td></td>
<td>Eingabeparameter</td>
<td></td>
</tr>
<tr>
<td></td>
<td>EndZeit :ZEITSTEMPEL_U TC</td>
<td>Zeit bis wann die Remote Service Bedienung aktiv ist. EndZeit==0 falls keine Servicebedienung aktiv ist.</td>
</tr>
<tr>
<td></td>
<td>ServiceGrund : string</td>
<td>Gibt an warum Service Bedienung angefordert wurde.</td>
</tr>
<tr>
<td></td>
<td>Ausgabeparameter</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RetCode : RetCode</td>
<td>OK: Entry wurde angelegt. PARAM_INVALID, INTERVALL_INVALID,ACCESS_DENIED</td>
</tr>
</tbody>
</table>
RemoteService
Pfad: ./ (nur eine Instanz pro Feldgerät)

<table>
<thead>
<tr>
<th>METHOD</th>
<th>Name</th>
<th>Beschreibung</th>
</tr>
</thead>
</table>

| Ausgabeparameter | RetCode : RetCode | OK, (auch wenn Remote Service Bedienung nicht aktiv war)ACCESS_DENIED |

4.1.4.1 Direkter externer Zugang zum Feldgerät

Direkte externe Zugänge zu den Feldgeräten über Wählerverbindungen/Netzwerke oder die lokale Serviceschnittstelle liegen in der Verantwortung des Feldgeräteherstellers. Sie werden projekt- bzw. herstellerspezifisch realisiert.

Falls zum Feldgerät neben der Verbindung zur Zentrale ein solcher direkter externer Zugang hergestellt wird, ist von OCIT-konformen Geräten folgendes zwingend zu erfüllen:

Beginn des Zugangs: Meldung WARTUNG_EIN zur Zentrale.

Ende des Zugangs: Meldung WARTUNG_AUS zur Zentrale.

Bei direkten Wählerverbindungen kann es dabei möglich sein, dass die Zentrale für die Zeit eines direkten Zuganges zum Feldgerät abgetrennt wird, und daher eine Verbindungsstörung erkennt. Dieser Störungszustand kann nach dem Beenden des Zugangs korrigiert werden, da eine entsprechende Meldung vom Feldgerät zur Zentrale erfolgt.

4.2 Meldungen und Messwerte (Archive)


4.2.1 Eigenschaften der Archive

In Archiven werden ausgewählte Daten und Meldungen des Feldgerätes gesammelt. Hierfür stellt OCIT folgende Elemente zur Verfügung (Details siehe Pkt. 4.2.9):

- Eine allgemeine Archivschnittstelle, welche die einfache Verwaltung von Archiven zulässt,
- Standard- und optionale Archive für Meldungen,
zur Laufzeit von der Zentrale definierbare Messwertarchive. Damit für Meldungen eine
definierte Speichertiefe eingehalten wird, sieht OCIT-Outstations für Messwerte eigene
Archive vor.

Die Archivschnittstelle für Meldungen und Messwerte hat folgende Vorteile:

- Es lassen sich Messwerte hinzufügen, ohne dass in OCIT-Outstations Erweiterungen ge-
macht werden müssen.
- Die Daten werden kompakt übertragen.
- Es ist möglich, herstellerabhängig zusätzliche Archive zu definieren.
- Messwerte können mehrfach abgeholt bzw. archiviert werden.
- Die möglichen, eintragbaren Meldungen im Archiv sind erweiterbar.
- Es ist möglich, herstellerabhängig einzelne Meldungen (durch neue Meldungsteile) zu
  erweitern.
- Trotz der Erweiterung einer Meldung durch neue Meldungsteile lässt sich die ursprüngli-
  che Meldung einfach herausfiltern.
- Durch Updates zur Zentrale bei neuen Meldungen bleibt die Zentrale auf dem neuesten
  Stand.
- Zentrale kann das gesamte, noch vorhandene Archiv lesen.
- Möglich ist eine inkrementelle Übertragung zu Zentrale1 parallel zu Gesamtlesen von
  Zentrale 2 (z.B. Systemzugang).
- Datenverlust erst bei Überlauf des Zwischenspeichers, nicht wegen Übertragungsstörung
  allein.
- Die Zentrale kann auf den jüngsten Eintrag zugreifen (für aktuellen Status bei Verbin-
  dungsauflauf).
- Die Meldungen lassen sich projektspezifisch auf Kundenwunsch ohne Firmwareänder-
  ungen anpassen.

4.2.2 Die Archivschnittstelle

Meldungen und Messwerte werden in einer gemeinsamen Schnittstelle behandelt. Die Daten-
strukturen und die definierten Funktionen der Schnittstelle sind für Meldungen und Messwer-
te strukturell gleich.

Meldungen und Messwerte werden in 'Listen' abgespeichert. Es gibt im Feldgerät mehrere
Listen, die unterschiedliche Daten abspeichern.Welche Daten abgespeichert werden, wird
durch 'Aufträge' festgelegt und in der jeweiligen Liste abgespeichert. Diese Konfiguration
kann während des Betriebs ausgelesen werden. Es gibt vordefinierte Listen, die vom Geräte-
hersteller fest definiert sind und dynamische Listen, die von der Zentrale während des Be-
triebs konfiguriert werden können. Pro Liste sind bis zu 256 verschiedene Aufträge möglich.

Jede Liste hat einen eigenen Puffer, in dem die dynamischen Daten gespeichert werden. Der
Puffer ist als Ringpuffer ausgeführt, in dem die jeweils ältesten Sätze überschrieben werden.
Die Größe des Ringpuffers ist konfigurierbar, kann aber während der Messwertaufnahme
nie geändert werden.

Der Ablauf sieht folgendermaßen aus: Das Feldgerät speichert neue Daten in seinem Ringpuffer, überträgt die Daten aber nicht automatisch zur Zentrale. Die Daten werden nur auf Anfrage von der Zentrale an diese übertragen. Damit ist es möglich, die Daten sekundengenau abzuspeichern, sie können jedoch in größeren Zeitabständen übertragen werden. Zusätzlich kann die Zentrale (mit Events) benachrichtigt werden, wenn der Puffer einen von der Zentrale festgelegten Füllgrad überschreitet. Zusätzlich können Events ausgelöst werden, wenn bestimmte Ereignisse eintreten (z.B. wenn eine Lampe ausgebrannt ist). Das Abholen der Daten geschieht über BTPPL-Methodencalls, die mindestens zusätzlich zu PPP und TCP über eine 16-Bit Prüfsumme (Fletcher) gesichert sind und optional über eine 160 Bit Prüfsumme (SHA-1) abgesichert werden können.

![Schema der Archivschnittstelle](image-url)
4.2.3 Elemente der Archivschnittstelle

Die Archiv-Schnittstelle ist in folgende abstrakte Elemente untergliedert:

- Eine **Liste** verwaltet Aufträge und die dazugehörigen dynamischen Werte, wie Messwerte oder Meldungen. Welche Werte erfasst und gespeichert werden, wird durch die Aufträge festgelegt. Es gibt statische Listen, die nicht neu definiert werden können und dynamische Listen, die von der Zentrale neu konfiguriert werden können.

- **Aufträge** gehören zu einer Liste und legen fest, welche dynamischen Werte gespeichert werden sollen. Ein Auftrag besteht aus einem oder mehreren Auftragselementen.

- **Auftragselemente** verweisen auf die Objekte (Prozessvariablen) und enthalten die Information, welche Werte dieser Objekte erfasst werden. Im Normalfall verweist ein Typ von Auftragselement genau auf einen Typ von OCIT-Outstations-Objekt, dessen Daten es erfasst. Es ist aber möglich, dass eine Prozessvariable von mehr als einem Auftragselement (in verschiedenen Listen) verwendet wird. Jedes Element in OCIT-Outstations, für das ein Auftragselement vorhanden ist, kann als **Datenquelle** eingesetzt werden.

- Der **Auftragsframe** ist das Ergebnis eines Auftrags. Auch ein Auftrag, der aus mehreren Auftragselementen besteht, erzeugt immer genau einen Auftragsframe.

- Die dynamischen Daten werden in **Sekundenframes** gespeichert. Ein Sekundenframe enthält z.B. eine Reihe von Messwerten in Form von Auftragsframes, die in der gleichen Sekunde entstanden sind, oder alle Meldungsteile, die zu einer Meldung gehören

- Ein spezieller Auftragsframe ist der **Meldungsauftrag** zur Erfassung von Meldungen. OCIT-konforme Meldungen bestehen aus einem **Hauptmeldungsteil** und 0..n **Zusatzmeldungsteilen** (optionalen Meldungsteilen). Der Hauptmeldungsteil bestimmt die Semantik der Meldung. Eine Meldung wird deshalb in Meldungsteile aufgeteilt, damit verschiedene Hersteller eine bereits vorhandene Standardmeldung erweitern können. Durch die Gruppierung wird dargestellt, dass die Meldungsteile semantisch zusammengehören. Ein **Meldungsteil** besteht aus einer Kennung und einem Satz von Parametern, welcher die Meldung genauer definieren. Die Kennung besteht aus der Herstellerkennung (Member-Nummer) und aus der Typkennung (OType).

<table>
<thead>
<tr>
<th>OType Nummer der Listen in der Zentrale, Member=0:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>OType</strong></td>
</tr>
<tr>
<td>----------</td>
</tr>
<tr>
<td>400</td>
</tr>
<tr>
<td>401</td>
</tr>
<tr>
<td>402</td>
</tr>
<tr>
<td>405</td>
</tr>
<tr>
<td>430</td>
</tr>
</tbody>
</table>

Alle Objekte mit Ausnahme des Events unterstützen die Standardfunktion ‚Get‘. Sie unterstützen nicht die Funktion ‚Set‘. Die zurückgelieferten Parameter werden in der XML-Datei genauer beschrieben.
4.2.4 Liste

Eine Liste hat zwei Aspekte: In einem statischen Teil wird abgelegt, welche Aufträge existieren und damit welche dynamischen Daten übertragen werden. Für diese Aufträge gibt es ein Prüfwort (Listenversion), welches bei jeder Änderung der Aufträge neu gesetzt wird und damit die Prüfung einfach macht, ob die übertragenen dynamischen Daten und die Auftragsversorgung kompatibel zueinander sind.

Der dynamische Teil der Liste (Ringpuffer) speichert die angefallenen Daten in sog. „Sekundenframes“. Diese Daten können dann von der Zentrale abgeholt werden. Es ist möglich, die Daten mehr als einmal abzuholen, da sie auch nach dem Abholen weiter gespeichert bleiben. Dadurch können z.B. auch Systemzugänge die aktuellen Messwerte verwenden. Es ist nicht vorgesehen, dass Systemzugänge Aufträge löschen oder sich als Eventziel beim Gerät eintragen (Gerät ist nicht Multimasterfähig). Die Daten werden in einem Ringpuffer gespeichert, bei welchem die alten Werte einfach überschrieben werden.

Zusätzlich gibt es die Möglichkeit für die Zentrale, sich einen Event schicken zu lassen, wenn ein vorher parametriert Füllgrad erreicht ist. Zusätzlich kann angegeben werden, wie viel Platz für den Ringpuffer einer Liste verwendet werden soll.

Die Listen sind fest adressiert (Pfad absolut zum Gerät, unabhängig von relativem Knoten) und übertragen Daten übergreifend zum relativen Knoten.

Jeder Sekundenframe hat einen Zeitstempel (sekundengenau) und eine Positionsnummer. Die Positionsnummer ist notwendig, um mehrere Elemente mit gleichem Zeitstempel voneinander zu unterscheiden. Die gleiche Positionsnummer darf im Ringpuffer der gleichen Liste nicht zweimal auftreten, kann aber „lückenhaft“ sein (auf 10 kann z.B. 50 folgen). Es ist im Gerät nicht notwendig, die Positionsnummern mit abzuspeichern. Stattdessen kann auch eine 32-Bit Speicheradresse oder eine Dateiposition verwendet werden. Der Ringpuffer muss selbstverständlich so groß sein, dass er mehr Elemente aufnehmen kann, als pro Sekunde maximal erzeugt werden können. (Das Tupel Zeit/Position darf pro Liste nur einmal vorkommen).

Pro Liste kann festgelegt werden, ob die Abfragen gesichert erfolgen müssen oder ungesichert sein können. Dies ist notwendig, da zum einen bei sehr großen Datenmengen (Onlinemesswerte) eine gesicherte Übertragung Systemressourcen benötigt, zum anderen aber bestimmte Listen sicherheitsrelevant sind.

Zusammenfassung:

- Das Prüfwort ermöglicht Konsistenzprüfung zwischen Listenkonfiguration und abgeholten Daten.
- Die Listeninstanzen sind vordefiniert, aber nicht aktiviert.
- Die Listen sind fest adressierbar (Pfad absolut zum Gerät, unabhängig von relativem Knoten).
- Die Liste verwaltet Datenframes.
- Die Daten werden über das Interface der Liste abgeholt.
- Zur Liste gehört ein Füllgrad, bei dem - optional - die Liste einen Event zur Zentrale schickt (Füllgrad in %).
- Die Liste hat eine feste Eigenschaft, wie viel Platz für die Frames reserviert ist.
- Festlegbar ist das Verhalten bei Überfüllung (Stoppen oder Überschreiben).

Jede Liste besitzt einen Ringpuffer von Sekundenframes, der abgefragt werden kann. Ein abgefragter Sekundenframe wird nicht gelöscht, sondern verbleibt im Ringpuffer, bis er überschrieben wird.

Jeder Sekundenframe hat einen Zeitstempel (sekundengenau) und eine (Ringpufferweit eindeutige) Positionsnummer.

In einem Sekundenframe sind die Daten von einem oder mehreren Aufträgen gespeichert.

Es ist möglich, dass mehrere Sekundenframes für die gleiche Sekunde existieren. Die Daten für einen Auftrag innerhalb derselben Sekunde sind immer innerhalb des gleichen Frames.

Es ist möglich, dass in einem Sekundenframe die gleiche Auftragsnummer mehrfach auftritt (speziell bei Meldungen).

Pro Liste kann festgelegt werden, ob die Abfragen gesichert erfolgen müssen oder ungesichert sein können.


Schema einer Liste

### 4.2.4.1 ÜbergabefORMAT

Das Feldgerät speichert nur Elemente chronologisch in dem Ringpuffer jeder Liste. Es speichert nicht, ob die Elemente schon von der Zentrale gelesen wurden.

Jeder Eintrag im Ringpuffer (Sekundenframe) hat einen Zeitstempel (UTC), also mit Sekundenaufflüssung. Es ist möglich, dass innerhalb einer Liste mehrere Elemente mit dem gleichen Zeitstempel existieren. Diese können im Fall einer Uhrzeitkorrektur auch Rückwärtsprüinge aufweisen. Uhrzeitkorrekturen meldet das Gerät mit der Betriebsmeldung (4.2.12) Zeitsprung. Damit kann die Zentrale den chronologischen Zusammenhang der Daten wiederherstellen.
Jeder Sekundenframe besteht aus einer Liste von Auftragsframes. Der Aufbau eines Auftragsframes hängt vom Typ (Member, OType) des Auftragsframes ab (Achtung: Diese Member/OType Kombination des Auftragsframes ist nicht die Meldungsnummer und wird nicht übertragen). Auftragsframes von Meldungen enthalten variable Parameter und unter anderem die Member/OType Kombination der Meldung (die auch übertragen wird).

Innerhalb einer Liste ist jeder Sekundenframe durch den Zeitstempel und eine Positionsnummer (innerhalb des Ringpuffers) eindeutig identifizierbar. Die Positionsnummer muss innerhalb des Ringpuffers eindeutig sein, sie kann aber Lücken haben. Denkbar ist als Positionsnummer beispielsweise ein File-Offset innerhalb einer Datei im Feldgerät oder die Speicheradresse des Elements im Speicher. Damit dies möglich ist, wird die Positionsnummer als ui4 festgelegt. Die Positionsnummer darf nur den Wertebereich 0x0 .. 0xffffffff umfassen, also kein 0xffffffff enthalten.

Wenn eine Liste von Sekundenframes vom Feldgerät in die Zentrale übertragen wird, überträgt man zusätzlich den Zeitstempel und die Positionsnummer des vorherigen Elements und den Zeitstempel und die Positionsnummer des letzten übertragenen Sekundenframes.

4.2.4.2 Methoden für Liste

Generell werden alle Methoden, die auf die Liste ausgeführt werden, authentifiziert. Die einzige Ausnahme ist GetSFSince, da GetSFSince keine Veränderung in der Liste selbst vornimmt. GetSFSinceEvent wird immer authentifiziert übertragen, bei der Antwort kann aber auf eine Authentifizierung verzichtet werden, sofern AuthenticateAnswer auf 0 gesetzt ist.

Faustregel: Wichtige Daten wie Betriebsmeldungen werden authentifiziert; große Datennugen, die nicht „gerichtsrelevant“ sind, wie Messwerte, werden zu Gunsten der Performance nicht authentifiziert.

In jeder Liste existiert implizit ein Auftrag mit der Nummer 0. Dieser fügt die folgenden Erignisse als Meldungen in den Ringpuffer der Liste ein:

Suspend, Unsuspend, Zeitsprung, StartAuftrag, StopAuftrag.

Interface des OBJTYPE Liste Member =0, OCIT Standard OType = 400 Liste

<table>
<thead>
<tr>
<th>METHOD</th>
<th>Name</th>
<th>Beschreibung</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>GetOldest</td>
<td>Ältestes Listenelement lesen</td>
</tr>
<tr>
<td></td>
<td>Ausgabeparameter</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RetCode : RetCode</td>
<td>OK: ältester Sekundenframe(SF) richtig geliefert</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NO_SF: Liste enthält überhaupt keinen Sekundenframe</td>
</tr>
<tr>
<td></td>
<td>PosNr : ULONG</td>
<td>Positionsnummer des gelieferten SF</td>
</tr>
<tr>
<td>METHOD</td>
<td>Name</td>
<td>Beschreibung</td>
</tr>
<tr>
<td>--------</td>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Listenversion: ui2</td>
<td>Versionskennung (Prüfwort), die bei jeder Auftragsänderung vom Feldgerät geändert wird. Die Versionsnummer wird auch beim Start der Liste zurückgegeben, so dass die Zentrale feststellen kann, ob eine Änderung stattgefunden hat. Wie unmittelbar aus den Datenstrukturen ersichtlich ist, ist die Versionsnummer der Liste nicht die Versionsnummer des Sekundenframes und als solche nicht in der Liste mit abgespeichert. Wenn also bei einem gestoppten Auftrag der Auftrag geändert wird, ist der Benutzer der Zentrale selbst für die Nebenwirkungen verantwortlich.</td>
<td></td>
</tr>
<tr>
<td>Sekundenframe</td>
<td>Ältester Sekundenframe (Achtung: Dieser Frame enthält eine Liste aus Auftragsframes!)</td>
<td></td>
</tr>
<tr>
<td>101</td>
<td>GetYoungest</td>
<td>Jüngstes Listenelement lesen</td>
</tr>
<tr>
<td>Ausgabeparameter</td>
<td>RetCode : RetCode</td>
<td>OK: jüngster Sekundenframe richtig geliefert</td>
</tr>
<tr>
<td></td>
<td></td>
<td>O_SF: Liste enthält überhaupt keine Sekundenframes</td>
</tr>
<tr>
<td></td>
<td>PosNr : ULONG</td>
<td>Positionsnummer des gelieferten SF</td>
</tr>
<tr>
<td>Listenversion: ui2</td>
<td>Versionskennung, die bei jeder Auftragsänderung vom Feldgerät geändert wird. Die Versionsnummer wird auch beim Start der Liste zurückgegeben, so dass die Zentrale feststellen kann, ob eine Änderung stattgefunden hat. Wie unmittelbar aus den Datenstrukturen ersichtlich ist, ist die Versionsnummer der Liste nicht die Versionsnummer des Sekundenframes und als solche nicht in der Liste mit abgespeichert. Wenn also bei einem gestoppten Auftrag der Auftrag geändert wird, ist der Benutzer der Zentrale selbst für die Nebenwirkungen verantwortlich.</td>
<td></td>
</tr>
<tr>
<td>Sekundenframe</td>
<td>Jüngster Sekundenframe (Achtung: Dieser Frame enthält eine Liste aus Auftragsframes!)</td>
<td></td>
</tr>
</tbody>
</table>
### Liste

<table>
<thead>
<tr>
<th>METHOD</th>
<th>Name</th>
<th>Beschreibung</th>
</tr>
</thead>
</table>

**Eingabeparameter**

- **Zeit**: ZEITSTEMPEL_UTC Zeitpunkt ab welchem Elemente gelesen werden
- **PosNr**: ui4 Positionsnummer ab der gelesen wird. Das erste zu liefernde Element ist das auf Zeit.PosNr folgende Element. Nullvalue=0xffffffff Die Positionsnummer 0xffffffff darf nicht auftreten.
- **MaxAnzahl**: Maximale Anzahl zu lesender Elemente

**Auszabeparameter**

- **RetCode**: RetCode SF_FOLLOW: Sekundenframes richtig geliefert und weitere Sekundenframes, die später als diese eingetragen wurden, in der Liste
  - SF_NOFOLLOW: Sekundenframes richtig geliefert und keine später eingetragenen Sekundenframes im Archiv.
  - NO_SF: Liste enthält keinen Sekundenframe, der die Bedingung erfüllt.
- **AbZeit**: ZEITSTEMPEL_UTC Zeitstempel des Sekundenframes, der direkt vor den übertragenen Sekundenframes im Ringpuffer eingetragen ist oder 0 falls kein solcher Sekundenframe im Ringpuffer ist.
- **AbPosNr**: ULONG Positionsnummer des Sekundenframes, der direkt vor den übertragenen Sekundenframes im Ringpuffer eingetragen ist oder undefiniert falls kein solcher Sekundenframe im Ringpuffer ist.

Wenn nacheinander mehrfach mit GetSFSince ausgelesen wird und keine Elemente überschrieben wurden, ist „AbZeit dieses Aufrufs“ = „Zeit des letzten Sekundenframes des letzten Aufrufs“ und „AbPosNr dieses Aufrufs“ = „PosNr des letzten Sekundenframes des letzten Aufrufs“. Existiert zum ersten gesendeten kein älteres Element mehr, ist AbPosNr undefiniert und AbZeit == 0.
<table>
<thead>
<tr>
<th>METHOD</th>
<th>Name</th>
<th>Beschreibung</th>
</tr>
</thead>
<tbody>
<tr>
<td>BisZeit:</td>
<td>ZEITSTEMPEL_UTC</td>
<td>Zeitstempel des letzten im folgenden gesendeten Elements also von Element[Anzahl-1].</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BisPosNr : ui4</td>
<td></td>
<td>Positionsnummer des letzten im folgenden gesendeten Elements also von Element[Anzahl-1].</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Listenversion: ui2</td>
<td></td>
<td>Versionskennung, die bei jeder Auftragsänderung vom Feldgerät geändert wird. Die Versionsnummer wird auch bei Auftragsänderungen zurückgegeben, so dass die Zentrale feststellen kann, ob eine Änderung stattgefunden hat. Wie unmittelbar aus den Datenstrukturen ersichtlich ist, ist die Versionsnummer der Liste nicht die Versionsnummer des Sekundenframes und als solche nicht in der Liste mit abgespeichert. Wenn also bei einem gestoppten Auftrag der Auftrag geändert wird, ist der Benutzer der Zentrale selbst für die Nebenwirkungen verantwortlich.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Anzahl : ui2</td>
<td></td>
<td>Anzahl der folgenden Sekundenframes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sekundenframes [] :</td>
<td></td>
<td>Gelesene Meldungen (Achtung: Jede Meldung enthält wieder eine Liste aus Meldungssteilen)</td>
</tr>
<tr>
<td>Sekundenframe</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**GetSFSinceWithEvent**

GetSFSinceWithEvent ist eine Kombination der Methoden GetSF-Since() und SetEvent(). Die Parameter LastTime und LastPosNr von SetEvent sind die des letzten von GetSF-Since zurückgegebenen Sekundenframes.

Diese Methode darf nur von dem als EventDestination eingetragenen Gerät (Zentrale) aus aufgerufen werden.


**Eingabeparameter**

| Zeit : utc                | Zeitpunkt ab welchem Elemente gelesen werden |
|                         |                                               |
| PosNr : ui4              | Positionsnummer ab der gelesen wird. Das erste zu liefernde Element ist das auf Zeit.PosNr folgende Element. |
| MaxAnzahl                | Maximale Anzahl zu lesender Elemente         |
| AuthenticateAnswer       | 0 (FALSE): Der Respond braucht nicht authentifiziert zu werden (keine SHA-1 Prüfsumme notwendig). Wenn trotzdem authentifiziert wird, muss die Prüfsumme stimmen. 1 (TRUE): Der Respond muss authentifiziert werden. |
| Ausgabeparameter         |                                               |
### Liste

<table>
<thead>
<tr>
<th>METHOD</th>
<th>Name</th>
<th>Beschreibung</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RetCode : ui2</td>
<td>SF_FOLLOW: Sekundenframes richtig geliefert und weitere Sekundenframes, die später als diese eingetragen wurden, in der Liste enthalten.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SF_NOFOLLOW: Sekundenframes richtig geliefert und keine später eingetragenen Sekundenframes im Archiv.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NO_EVENT, wenn aus irgendeinem Grund der Event nicht eingetragen werden kann.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NO_SF: Liste enthält keinen Sekundenframe, der die Bedingung erfüllt.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ACCESS_DENIED Zugriff nicht erlaubt, da nicht von Event-Destination ausgelöst.</td>
</tr>
<tr>
<td></td>
<td>AbZeit : utc</td>
<td>Zeitstempel des Sekundenframes, der direkt vor den übertragenen Sekundenframes im Ringpuffer eingetragen ist oder 0 falls kein solcher Sekundenframe im Ringpuffer ist.</td>
</tr>
<tr>
<td></td>
<td>AbPosNr : ui4</td>
<td>Positionsnummer des Sekundenframes, der direkt vor den übertragenen Sekundenframes im Ringpuffer eingetragen ist oder undefiniert falls kein solcher Sekundenframe im Ringpuffer ist.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Wenn nacheinander mehrfach mit GetSFSince ausgelesen wird und keine Elemente überschrieben wurden, ist 'AbZeit dieses Aufrufs' = 'Zeit des letzten Sekundenframes des letzten Aufrufs' und 'AbPosNr dieses Aufrufs' = 'PosNr des letzten Sekundenframes des letzten Aufrufs'.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Existiert zum ersten gesendeten kein älteres Element mehr, ist AbPosNr undefiniert und AbZeit == 0.</td>
</tr>
<tr>
<td></td>
<td>BisZeit:</td>
<td>ZEITSTEMPEL.UTC Zeitstempel des letzten im folgenden gesendeten Elements also von Element[Anzahl-1].</td>
</tr>
<tr>
<td></td>
<td>BisPosNr : ui4</td>
<td>Positionsnummer des letzten im folgenden gesendeten Elements also von Element[Anzahl-1].</td>
</tr>
<tr>
<td></td>
<td>Listenversion: ui2</td>
<td>Versionskennung, die bei jeder Auftragsänderung vom Feldgerät geändert wird. Die Versionsnummer wird auch bei Auftragsänderungen zurückgegeben, so dass die Zentrale feststellen kann, ob eine Änderung stattgefunden hat.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Wie unmittelbar aus den Datenstrukturen ersichtlich ist, ist die Versionsnummer der Liste nicht die Versionsnummer des Sekundenframes und als solche nicht in der Liste mit abgespeichert. Wenn also bei einem gestoppten Auftrag der Auftrag geändert wird, ist der Benutzer der Zentrale selbst für die Nebenwirkungen verantwortlich.</td>
</tr>
<tr>
<td></td>
<td>Anzahl : ui2</td>
<td>Anzahl der folgenden Sekundenframes</td>
</tr>
<tr>
<td></td>
<td>Sekundenframes [] : Sekundenframe</td>
<td>Gelesene Meldungen (Achtung: Jede Meldung enthält wieder eine Liste aus Meldungsteilen)</td>
</tr>
</tbody>
</table>
### Liste

<table>
<thead>
<tr>
<th>METHOD</th>
<th>Name</th>
<th>Beschreibung</th>
</tr>
</thead>
<tbody>
<tr>
<td>104</td>
<td>SetEvent</td>
<td>Beauftragt die Liste beim überschreiten des angegebenen Füllgrads „Fill“ die Methode EvList::OnFull() im durch SetEventDestination() angegebenen Gerät (in der Regel die Zentrale) aufzurufen. Die Übergabeparameter LastTime und LastPosNr markieren die Startposition zur Ermittelung des aktuellen Füllgrads. Jedes mal wenn nach dem Eintragen eines Sekundenframes der Füllgrad überschritten wird, wird die Methode OnFull (0, ArchivEvent-Objekt ) aufgerufen. Wenn beim Füllgrad ein Wert &gt; 100 eingetragen wird, wird kein neues Event ausgelöst. Bei einem Füllgrad von 0 wird das Event nach jedem neuen Eintrag ausgelöst.</td>
</tr>
</tbody>
</table>

#### Eingabeparameter

- **LastTime**: Zeitpunkt des letzten Elements, das abgeholt wurde
- **LastPosNr**: Positionsnummer des letzten Elements, das abgeholt wurde
- **Fill**: Maximaler Füllgrad, bei dem EvList::OnFull ausgelöst wird.  
  - Fill=0 ➔ nach jedem Eintrag OnFull auslösen  
  - Fill>100 ➔ nie OnFull auslösen

#### Ausgabeparameter

- **RetCode**:  
  - ui2: OK, wenn das Event eingetragen werden kann;  
  - NO_EVENT, wenn aus irgendeinem Grund das Event nicht eingetragen werden kann  
  - ACCESS_DENIED Zugriff nicht erlaubt, da nicht von Event-Destination ausgelöst.

| 105    | Start | Startet die Liste, d.h. die Aufträge werden scharf. Auch die Aggregation von AEAggregiert startet. Wenn die Liste gestartet wird, wird der Ringpuffer gelöscht. Es ist bei bestimmten Listen herstellerabhängig möglich, dass sie nicht gestoppt (und gestartet) werden können. |

#### Eingabeparameter

- Keiner

#### Ausgabeparameter

- **RetCode**:  
  - OCIT_OK : Datenerfassung ist gestartet  
  - NOT_POSSIBLE: Liste kann prinzipiell nicht gestartet werden.  
  - NOT_INACTIVE: Datenerfassung ist bereits gestartet (Liste wurde nicht gelöscht)  
  - ERROR: Befehl nicht ausführbar (Auftraege unvollständig konfiguriert)
<table>
<thead>
<tr>
<th>METHOD</th>
<th>Name</th>
<th>Beschreibung</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Listenversion</td>
<td>Versionsnummer, die auch bei GetSFSinceXXX zurückgegeben wird.</td>
</tr>
<tr>
<td>106</td>
<td>Stop</td>
<td>Hält die Datenerfassung der Liste an. Es ist nach dem Stop der Liste noch möglich, den Ringpuffer abzuholen. Es ist bei bestimmten Listen herstellerabhängig möglich, dass sie nicht gestoppt (und gestartet) werden können.</td>
</tr>
<tr>
<td></td>
<td>Eingabeparameter</td>
<td>Keiner</td>
</tr>
<tr>
<td></td>
<td>Ausgabeparameter</td>
<td>RetCode</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OCIT_OK : Liste ist gestoppt, wird auch bei einer bereits gestoppten Liste zurückgegeben.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NOT_POSSIBLE : Liste kann prinzipiell nicht gestoppt werden.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ERROR : Befehl konnte nicht ausgeführt werden.</td>
</tr>
<tr>
<td></td>
<td>Eingabeparameter</td>
<td>Keine</td>
</tr>
<tr>
<td></td>
<td>Ausgabeparameter</td>
<td>RetCode</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OCIT_OK</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RetCode != OCIT_OK -&gt; Befehl konnte nicht ausgeführt werden.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ListenversionAlt, ListenversionNeu</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Listenversion vor Reset</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Listenversion nach Reset</td>
</tr>
<tr>
<td></td>
<td>Eingabeparameter</td>
<td>Member : USHORT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OType : USHORT</td>
</tr>
<tr>
<td>METHOD</td>
<td>Name</td>
<td>Beschreibung</td>
</tr>
<tr>
<td>------------</td>
<td>--------------------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>RetCode</td>
<td>OCIT_OK</td>
<td>PARAM_INVALID mit Member/OType angegebener Auftrags-typ ist nicht bekannt, Auftrag wurde nicht angelegt.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NOT_INACTIVE der Auftrag darf nicht gestartet sein um diese Methode auszuführen</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BUFFER_TOO_SMALL: wird geliefert, wenn der Sekunden-frame so groß werden kann, dass weniger als vier Einträge in den Ringpuffer passen.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NOT_POSSIBLE: Anzahl Aufträge zu groß (max. 255 Aufträge möglich)</td>
</tr>
<tr>
<td>AuftragsNr : UBYTE</td>
<td>Auftragsnummer des neu angefügten Auftrags falls RetCode ==OK. Der neue Auftrag kann mit dem Pfad Liste()/AuftragsNr() angesprochen werden.</td>
<td></td>
</tr>
<tr>
<td>ListenversionAlt, ListenversionNeu</td>
<td>Listenversion vor AddAuftrag</td>
<td>Listenversion nach AddAuftrag</td>
</tr>
<tr>
<td>Eingabeparameter</td>
<td>ZNr, FNr : USHORT</td>
<td>ZNr/FNr des Gerätes, welches die Traps empfangen soll.</td>
</tr>
<tr>
<td>Ausgabeparameter</td>
<td>RetCode</td>
<td>RetCode != OCIT_OK -&gt; Befehl konnte nicht ausgeführt werden</td>
</tr>
<tr>
<td>ListenversionAlt, ListenversionNeu</td>
<td>Listenversion vor SetEventDestination</td>
<td>Listenversion nach SetEventDestination</td>
</tr>
<tr>
<td>Eingabeparameter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>METHOD</td>
<td>Name</td>
<td>Beschreibung</td>
</tr>
<tr>
<td>--------------</td>
<td>------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Persistenz :</td>
<td>UBYTE</td>
<td>Gibt an welche Teile der Liste nach Netzausfall erhalten bleiben: Keine, die gesamte Liste ist nach Netzausfall auf Standardwerte zurückgesetzt.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Aufträge, die Listenaufträge bleiben erhalten, die Daten (Inhalt des Ringpuffers) gehen verloren. Nach Netzausfall wird die Liste automatisch wieder gefüllt.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Alles, sowohl die Listenaufträge als auch der Inhalt des Ringpuffers bleiben über einen Netzausfall erhalten.</td>
</tr>
<tr>
<td>ListeSizeP  :</td>
<td>UBYTE</td>
<td>Minimale gewünschte Größe der Liste in Prozent des restlich verfügbaren Speichers. Es wird die Größe genommen, die zu mehr Bytes führt, aber nicht mehr als 100%. Die anderen Listen werden nicht verkleinert.</td>
</tr>
<tr>
<td>Ausgabeparameter</td>
<td>RetCode</td>
<td>OCIT_OK: Der Befehl wurde ausgeführt</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NOT_INACTIVE: Die Liste darf nicht gestartet sein.</td>
</tr>
<tr>
<td></td>
<td>CurrentSizeB :  ULONG</td>
<td>Größe der Liste in Byte, die eingestellt wurde, bzw. Größe der aktuellen Liste, falls keine Änderung möglich ist.</td>
</tr>
<tr>
<td></td>
<td>CurrentSizeP :  UBYTE</td>
<td>Größe der Liste in Prozent des vor dem Aufruf verfügbaren Speichers.</td>
</tr>
<tr>
<td></td>
<td>ListenversionAlt, ListenversionNeu</td>
<td>Listenversion vor SetSize Listenversion nach SetSize</td>
</tr>
<tr>
<td></td>
<td>SetOverwriteOnFull</td>
<td>Legt das Verhalten bei vollem Ringpuffer fest.</td>
</tr>
<tr>
<td></td>
<td>OverwriteOnFull : BOOL</td>
<td>Falls gesetzt (true), überschreibt die Liste die alten Datenframes (Ringpuffer). Andernfalls stoppt die Liste. Die Defaulteinstellung nach Reset ist true (Ringpuffer).</td>
</tr>
<tr>
<td>Ausgabeparameter</td>
<td>RetCode</td>
<td>OK, NOT_POSSIBLE falls das Gerät diese Einstellung nicht unterstützt.</td>
</tr>
<tr>
<td></td>
<td>Suspend</td>
<td>Hält die Erfassung der Liste an, ohne sie richtig zu stoppen. Der Ringpuffer bleibt erhalten. Wenn die Liste suspendiert wurde, wird die Betriebsmeldung Suspend in der Liste gespeichert.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Keine</td>
</tr>
<tr>
<td>Ausgabeparameter</td>
<td></td>
<td>OCIT_OK: Der Befehl wurde ausgeführt.</td>
</tr>
<tr>
<td>METHOD</td>
<td>Name</td>
<td>Beschreibung</td>
</tr>
<tr>
<td>--------</td>
<td>------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>113</td>
<td>Unsuspend</td>
<td>Wenn die Liste mit Supend angehalten ist, wird dieses „Suspend“ wieder zurückgenommen und die Betriebsmeldung Unsuspend in den Ringpuffer der Liste gespeichert.</td>
</tr>
</tbody>
</table>

### 4.2.4.3 Eventhandler für Listen in der Zentrale

Für alle Listen aller Feldgeräte existiert in der Zentrale eine Instanz des OBJTYPE EvList. Ein Event ist gleich, wenn Ziel (Zentrale), Methode und Eingabeparameter übereinstimmen.

Das Feldgerät sendet das nächste Event erst nachdem es eine Quittung auf das vorangegangene gleiche Event erhalten hat oder wenn es erneut aktiviert wurde (mit Liste::SetEvent(), Liste::GetSFSinceWithEvent(), Auftrag::ActivateEvent()). Dies ermöglicht eine Flusskontrolle für Events unter Berücksichtigung der Übertragungsdauer der Übertragungsstrecke sowie der Verarbeitung der Events durch die Zentrale. Die Übertragungsgeschwindigkeit der Events passt sich an die vorhandene Bandbreite und das aktuelle Übertragungsvolumen an.

Die Eventhandler werden folgendermaßen aufgerufen:

<table>
<thead>
<tr>
<th>METHOD</th>
<th>Name</th>
<th>Beschreibung</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>OnFull</td>
<td>Wird vom Feldgerät in der EventDestination (meist gleich Zentrale) aufgerufen, wenn der Füllgrad überschritten ist.</td>
</tr>
<tr>
<td></td>
<td>Eingabeparameter</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ZNr: USHORT</td>
<td>Absender Feldgerät</td>
</tr>
<tr>
<td></td>
<td>FNr: USHORT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Liste: UBYTE</td>
<td>Listennummer der Liste deren Füllstand überschritten wurde.</td>
</tr>
<tr>
<td></td>
<td>Ausgabeparameter</td>
<td></td>
</tr>
<tr>
<td>201</td>
<td>OnInvalidate</td>
<td>Wird vom Feldgerät in der EventDestination (meist gleich Zentrale) aufgerufen, wenn ein anderes Event-Ziel gesetzt wird. Siehe Liste::SetEventDestination().</td>
</tr>
<tr>
<td></td>
<td>Eingabeparameter</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ZNr:USHORT,</td>
<td>Auslösendes Feldgerät</td>
</tr>
<tr>
<td></td>
<td>FNr:USHORT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ListenNr : UBYTE</td>
<td>Liste deren Event neu gesetzt wurde.</td>
</tr>
<tr>
<td></td>
<td>ZNrNeu:USHORT,</td>
<td>Gerätenummer, des Gerätes welches das neue Event beantragt hat.</td>
</tr>
<tr>
<td></td>
<td>FNrNeu:USHORT</td>
<td></td>
</tr>
<tr>
<td>METHOD</td>
<td>Name</td>
<td>Beschreibung</td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
<td>---------------</td>
</tr>
<tr>
<td></td>
<td>Ausgabeparameter</td>
<td></td>
</tr>
<tr>
<td>202</td>
<td>OnInsert</td>
<td>Wird vom Feldgerät in der EventDestination (meist gleich Zentrale) aufgerufen, wenn die dynamischen Daten eines ‚aktivierten’ Auftrags eingetragen wurden. Siehe z.B. ActivateEvent</td>
</tr>
<tr>
<td></td>
<td>Eingabeparameter</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ZNr:USHORT,</td>
<td>Auslösendes Feldgerät</td>
</tr>
<tr>
<td></td>
<td>FNr:USHORT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ListenNr : UBYTE</td>
<td>Liste deren Event neu gesetzt wurde.</td>
</tr>
<tr>
<td></td>
<td>AuftragNr : UBYTE</td>
<td>Auftrag, der das Event ausgelöst hat.</td>
</tr>
<tr>
<td></td>
<td>Ausgabeparameter</td>
<td></td>
</tr>
<tr>
<td>203</td>
<td>OnNetzAus</td>
<td>wird vom Feldgerät (meist gleich Zentrale) aufgerufen, wenn ein Netzausfall erkannt wird</td>
</tr>
<tr>
<td></td>
<td>Eingabeparameter</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ZNr:USHORT,</td>
<td>Auslösendes Feldgerät</td>
</tr>
<tr>
<td></td>
<td>FNr:USHORT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Vorgangskennung:SYSJOBID</td>
<td>Vorgangskennung der Störung, identisch mit Vorgangskennung der korrespondierenden NetzAus-Meldung im Standard-Meldearchiv</td>
</tr>
<tr>
<td></td>
<td>Ausgabeparameter</td>
<td></td>
</tr>
</tbody>
</table>

4.2.5 Sekundenframe / Auftragsframe

Jeder Sekundenframe besteht aus einem oder mehreren Auftragsframes. Ein Auftragsframe besteht aus einer Auftragsnummer (1 Byte) sowie den dynamischen Daten, die der Auftrag generiert hat. Diese Daten beinhalten bei Messwerten im Normalfall nur die reinen Sachdaten, bei Meldungen wird zusätzlich die Member/OType-Kennung gesendet, die die ‚Meldungsnummer’ des Meldungssteils repräsentiert. Zu jedem Meldungssteil kann der Hersteller in der TYPE-Datei optional einen Formatstring angeben, mit dessen Hilfe die Zentrale den Mel-
dungsteil in ein lesbares Format übersetzt. Fehlt der Formatstring, gibt die Zentrale die Daten in einem beliebigen Format aus.

Bei Meldungen folgt auf die Member/OType-Kennung der Parametersatz, der die Meldung genauer beschreibt. Der Aufbau und die Länge der Parametersätze ist von der Member/OType-Kennung abhängig. Um unbekannte Meldungsteile überspringen zu können, wird bei jedem Meldungsteil nach dem Member/OType-Feld eine 2Byte Länge des Parametersatzes gespeichert.

Schema Sekunden- und Auftragsframe

4.2.6 Auftrag

Zu jeder Liste gehört eine Reihe von Aufträgen die festlegen, welche dynamischen Daten in der Liste abgespeichert werden. Jeder Auftrag gehört zu genau einer Liste, listenübergreifende Aufträge sind nicht vorgesehen.

Innerhalb einer Liste ist ein Auftrag eindeutig durch ein UBYTE (Auftragsnummer) identifiziert. Der Auftrag ist damit durch den Pfad Geraet/Liste()/Auftrag() zu erreichen. Bei dynamischen Listen ist ein Auftrag nicht von vornherein vorhanden, sondern muss vorher


Bestimmte Auftragstypen können sich aus Auftragselementen zusammensetzen, in denen dann gespeichert wird, welche Daten übertragen werden sollen. Andere Auftragstypen enthalten bereits implizit die Daten, die dort übertragen werden, z.B. der Auftragstyp für Meldungen.

- Es gibt unterschiedliche Typen von Aufträgen
- Alle Aufträge sind ObjTypes unter der jeweiligen Liste (Pfad: Liste()/Auftrag())
- Ein Auftrag aggregiert mehrere Auftragselemente.

Jeder Auftrag enthält implizit eine Bedingung. Wenn die Bedingung erfüllt ist, werden die dynamischen Daten des Auftrags erfasst und in einem Sekundenframe des Ringpuffers gespeichert.


Beim Start einer Liste und beim Start eines Auftrags während des Laufes einer Liste wird immer eine (neue) Listenversion zurückgegeben.

Um sicherzustellen, dass keine Terminals die Aufträge von Listen ändern können, gilt: Das Feldgerät weist alle Änderungsaufträge zurück, die nicht von der EventDestination stammen.

### 4.2.6.1 Methoden aller Aufträge

<table>
<thead>
<tr>
<th>METHOD</th>
<th>Name</th>
<th>Beschreibung</th>
</tr>
</thead>
</table>

OCIT-O-Basis_V1.1_A02 Copyright © 2004 ODG Seite 38 von 57
## Auftrag

<table>
<thead>
<tr>
<th>METHOD</th>
<th>Name</th>
<th>Beschreibung</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Eingabeparameter</td>
<td>Member : USHORT Auftragselement, welches hinzuzufügen ist</td>
</tr>
<tr>
<td></td>
<td>Ausgabeparameter</td>
<td>RetCode OK : wird zurückgeliefert, wenn das Auftragselement hinzugefügt werden konnte</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NOT_INACTIVE: wird zurückgegeben, wenn zwar theoretisch Auftragselemente hinzugefügt werden könnten, aber der Auftrag aktiv ist.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AENr : UBYTE Nummer des Auftragselements, das hinzugefügt wurde. Bei != OK ist der Wert undefiniert.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ListenversionAlt, ListenversionNeu Listenversion vor AddElement</td>
</tr>
<tr>
<td></td>
<td>Eingabeparameter</td>
<td>Keine</td>
</tr>
<tr>
<td></td>
<td>Ausgabeparameter</td>
<td>RetCode OK: wird zurückgeliefert, wenn der Auftrag erfolgreich aktiviert wurde. NOT_POSSIBLE : wird geliefert, wenn der Auftragstyp (bzw. die Liste) nicht zulässt, dass der Auftrag gestartet werden kann. NOT_ACTIVE : wird geliefert, wenn die Liste inaktiv ist.</td>
</tr>
</tbody>
</table>
### Auftrag

<table>
<thead>
<tr>
<th>METHOD</th>
<th>Name</th>
<th>Beschreibung</th>
</tr>
</thead>
<tbody>
<tr>
<td>122</td>
<td>Stop</td>
<td>Stoppt den Auftrag, wenn die Liste aktiv ist (und es möglich ist, den Auftrag getrennt zu stoppen). Wenn die Liste nicht aktiv ist, wird ein Fehler zurückgegeben. Stop löst einen Meldungseintrag StopAuftrag aus, der ggf. in die Liste selbst eingetragen wird. Diese Methode wird in OCIT-Outstations Version 1 nicht realisiert.</td>
</tr>
</tbody>
</table>

#### Eingabeparameter
Keine

#### Ausgabeparameter

| RetCode | OK: wird zurückgeliefert, wenn der Auftrag erfolgreich aktiviert wurde. NOT_POSSIBLE: wird geliefert, wenn der Auftragstyp (bzw. die Liste) nicht zulässt, dass der Auftrag gestoppt werden kann. NOT_ACTIVE: wird geliefert, wenn die Liste inaktiv ist. |

### 4.2.6.2 Meldungsauftrag


Bei Meldungsaufträgen, die standardmäßig Bestandteil von Archiven sind, ist die Einordnung von Meldungen in die Include- bzw. Exclude-Liste archivspezifisch vorgegeben.

Bei Meldungsaufträgen, die eine Zentrale anlegt, gelten folgende Festlegungen:

- Ein Meldungsauftrag besitzt nach dem Anlegen durch die Zentrale den Degree I
- Alle Meldungen befinden sich in der Exclude-Liste, d.h. standardmäßig generiert der Auftrag keine Meldungsframes. Nach ResetMT werden alle Meldungen, die dem Degree des Auftrags entsprechen, in die Include-Liste aufgenommen.


### Interface der Meldungsaufträge

<p>| Meldungsauftrag |</p>
<table>
<thead>
<tr>
<th>METHOD</th>
<th>Name</th>
<th>Beschreibung</th>
</tr>
</thead>
<tbody>
<tr>
<td>120,</td>
<td>AddElement,</td>
<td>Siehe 4.2.6</td>
</tr>
<tr>
<td>122</td>
<td>Stop</td>
<td></td>
</tr>
<tr>
<td>130</td>
<td>IncludeMT</td>
<td>Wird aufgerufen, wenn ein Haupt-Meldungsteil aus diesem Meldungsdegree (und damit dem Meldungsauftrag) entfernt werden soll. Es ist damit möglich, dass ein Haupt-Meldungsteil nicht mehr übertragen wird. Wenn der Eintrag in der Include-Liste bereits vorhanden ist, wird er aus der Include-Liste entfernt.</td>
</tr>
<tr>
<td></td>
<td>Eingabeparameter</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Member : USHORT</td>
<td>Haupt-Meldungsteil</td>
</tr>
<tr>
<td></td>
<td>OType : USHORT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ausgabeparameter</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RetCode OK:</td>
<td>wird zurückgeliefert, wenn der Meldungsteil erfolgreich eingetragen wurde.</td>
</tr>
<tr>
<td></td>
<td>NOT_POSSIBLE :</td>
<td>wird geliefert, wenn in dieser Liste Meldungsteile nicht ein- oder ausgehängt werden können.</td>
</tr>
<tr>
<td></td>
<td>PARAM_INVALID :</td>
<td>der angegebene Meldungsteil ist kein dem Gerät bekannter Haupt-Meldungsteil (oder ein ganz anderes Objekt).</td>
</tr>
<tr>
<td></td>
<td>NOT_INACTIVE :</td>
<td>Obwohl die Änderungen während des Laufes unproblematisch sind, kann herstellerabhängig (und ggf. projektabhängig) gefordert werden, dass in einer aktiven Liste keine Änderungen im In- und Exclude gemacht werden. In diesem Fall wird NOT_INACTIVE zurückgegeben.</td>
</tr>
<tr>
<td></td>
<td>ListenversionAlt,</td>
<td>Listenversion vor IncludeMT</td>
</tr>
<tr>
<td></td>
<td>ListenversionNeu</td>
<td>Listenversion nach IncludeMT</td>
</tr>
<tr>
<td></td>
<td>Eingabeparameter</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Member : USHORT</td>
<td>Haupt-Meldungsteil</td>
</tr>
<tr>
<td></td>
<td>OType : USHORT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ausgabeparameter</td>
<td></td>
</tr>
</tbody>
</table>
**Meldungsauftrag**

<table>
<thead>
<tr>
<th>METHOD</th>
<th>Name</th>
<th>Beschreibung</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ListenversionAlt,</td>
<td>Listenversion vor ExcludeMT</td>
</tr>
<tr>
<td></td>
<td>ListenversionNeu</td>
<td>Listenversion nach ExcludeMT</td>
</tr>
<tr>
<td>132</td>
<td><strong>ResetMT</strong></td>
<td>Die Include und Exclude-Liste wird auf geräteinterne Defaultwerte zurückgesetzt.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Eingabeparameter</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Keine</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ausgabeparameter</td>
</tr>
<tr>
<td></td>
<td>RetCode</td>
<td>OK: wird zurückgeliefert, wenn der Meldungsteil erfolgreich eingetragen wurde. NOT_POSSIBLE: wird geliefert, wenn in dieser Liste Meldungsteile nicht ein- oder ausgehängt werden können.</td>
</tr>
<tr>
<td></td>
<td>ListenversionAlt,</td>
<td>Listenversion vor ResetMT</td>
</tr>
<tr>
<td></td>
<td>ListenversionNeu</td>
<td>Listenversion nach ResetMT</td>
</tr>
<tr>
<td>133</td>
<td><strong>GetInEx</strong></td>
<td>Die Include und Exclude-Liste wird zurückgegeben. Alle vom Gerät generierten Meldungen befinden sich entweder in der Include- oder Exclude-Liste des Auftrags.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Eingabeparameter</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Keine</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ausgabeparameter</td>
</tr>
<tr>
<td></td>
<td>IncludeAnzahl: USHORT</td>
<td>Liste der aktuellen Include Meldungsteile.</td>
</tr>
<tr>
<td></td>
<td>IncludeMT[]: {Member,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>OType}</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ExcludeAnzahl: USHORT</td>
<td>Liste der aktuellen Exclude Meldungsteile.</td>
</tr>
<tr>
<td></td>
<td>ExcludeMT[]: {Member,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>OType}</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RetCode</td>
<td>OK</td>
</tr>
<tr>
<td>119</td>
<td><strong>ActivateEvent</strong></td>
<td>Aktiviert bzw. deaktiviert den Event, der ausgelöst wird, wenn in die Liste ein Element dieses Auftrags kommt.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Eingabeparameter</td>
</tr>
</tbody>
</table>
### Meldungsauftrag

<table>
<thead>
<tr>
<th>METHOD</th>
<th>Name</th>
<th>Beschreibung</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ActivateIt : UBYTE</td>
<td>0: Der Event wird deaktiviert</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: Der Event wird aktiviert</td>
</tr>
<tr>
<td>Ausgabeparameter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RetCode</td>
<td></td>
<td>OK: Der Event wurde erfolgreich aktiviert bzw. deaktiviert.</td>
</tr>
<tr>
<td>ListenversionAlt, ListenversionNeu</td>
<td></td>
<td>Listenversion vor ActivateEvent</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Listenversion nach ActivateEvent</td>
</tr>
<tr>
<td>134</td>
<td>SetDegree</td>
<td>Versieht den Meldungsauftrag mit einem neuen Meldungsg-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>degree. Die Include-/Exclude-Liste wird nicht verändert.</td>
</tr>
<tr>
<td>Eingabeparameter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Meldungsdegree : UByte</td>
<td>Neuer Meldungsdegree</td>
<td></td>
</tr>
<tr>
<td>Ausgabeparameter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RetCode</td>
<td></td>
<td>OK: Der Meldungsdegree wurde erfolgreich zugewiesen</td>
</tr>
<tr>
<td>ListenversionAlt, ListenversionNeu</td>
<td></td>
<td>Listenversion vor SetDegree</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Listenversion nach SetDegree</td>
</tr>
</tbody>
</table>

### 4.2.7 Meldung


4.2.8 Meldungsteil

Jeder Meldungsteil wird in einem speziellen Auftragssegment gespeichert. Pro Meldungsteil wird eine Auftragsnummer gespeichert, die für die Meldung selbst nicht relevant ist. Auf die Auftragsnummer folgt die Anzahl der Meldungsteile und die Member/OType-Kennung, welche die Meldungsnummer angibt. Die darauf folgende Länge des Parameterblocks ist sinnvoll, falls ein Meldungsteil aus irgendwelchen Gründen in der XML-Datei nicht bekannt ist. In einem solchen Fall könnten alle folgenden Meldungsteile und auch alle folgenden Sekundenframes/Meldungen nicht mehr ausgewertet werden. Der Mindestwert der Länge ist 4 (Länge der Vorgangskennung).

Der Aufbau der Vorgangskennung ist in 2.3 Vorgangskennung, SYSJOBID beschrieben. Es ist nicht notwendig, dass jeder Meldungsteil eine Vorgangskennung bekommt; in solchen Fällen wird eine 0 übertragen. Der Vorteil eines solchen Feldes liegt darin, dass die Zentrale in der Lage ist, bei unterschiedlichsten Meldungstypen in jedem Fall die Einträge zu finden, die zu einem bestimmten Vorgang gehören.
Die Parameter sind pro Meldung unterschiedlich und werden in der XML-Datei festgelegt, so dass sie in der Zentrale ausgewertet werden können.

Es gibt zwei verschiedene Typen von Meldungsteilen: Haupt-Meldungsteile und Zusatzmeldungensteile. Eine „Meldung“ besteht immer aus genau einem Hauptmeldungsteil und 0..n Zusatzmeldungsteilen.

4.2.8.1 Kategorie und Schweregrad eines Meldungsteils

OCIT-Outstations definiert eine Reihe Kategorien und Schweregrade für Meldungen. Die Kategorie und der Schweregrad der Meldung sind nur am Hauptmeldungsteil festgemacht, die Zusatzmeldungsteile haben weder eine Kategorie noch einen Schweregrad.

Diese Kategorien und Schweregrade sind an zwei Punkten definiert: In der XML-Datei wird die Default-Kategorie und der Default-Schweregrad der Meldung gespeichert. Wenn projekt-spezifisch die Kategorie und der Schweregrad geändert werden soll, so muss das im Feldgerät zusätzlich eingetragen werden. Dies geschieht über die Methoden IncludeMT und ExcludeMT aus der Zentrale heraus.

Folgende Kategorien und Schweregrade sind definiert:

<table>
<thead>
<tr>
<th>MeldungsDegree</th>
<th>Beschreibung</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information (0)</td>
<td>Hat keine Auswirkung auf den Verkehr.</td>
</tr>
<tr>
<td>Warnung (1)</td>
<td>Hat keine Auswirkungen auf den Verkehr, sollte aber bearbeitet werden.</td>
</tr>
<tr>
<td>Fehler (2)</td>
<td>Hat keine wesentliche Auswirkung auf den Verkehr.</td>
</tr>
<tr>
<td>Schwerer Fehler (3)</td>
<td>Hat wesentliche Auswirkung auf den Verkehr.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Meldungskategorie</th>
<th>Beschreibung</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sonstige (0)</td>
<td>Keine der folgenden Kategorien.</td>
</tr>
<tr>
<td>Geräte HW (1)</td>
<td>Geräte Hardware allgemein.</td>
</tr>
<tr>
<td>Sollbildfehler (2)</td>
<td>Signalsicherung: Steuerung versuchte fehlerhaftes Bild zu stellen.</td>
</tr>
<tr>
<td>Istbildfehler (3)</td>
<td>Signalsicherung: Lampenausfall in Signalgebern / Tastern, Ausbleiben oder ungewolltes Erscheinen eines Bildes.</td>
</tr>
<tr>
<td>VA (4)</td>
<td>Verkehrstechnische Meldungen in Anwenderprogrammen.</td>
</tr>
<tr>
<td>Übertragungssystem (5)</td>
<td>Kommunikation zur Zentrale (es ist zu erwarten, dass schwere Fehler dieser Kategorie nicht sofort nach oben gesendet werden).</td>
</tr>
<tr>
<td>Betriebssystem, Firmware (6)</td>
<td>Systemfehler und sonstiges.</td>
</tr>
<tr>
<td>Anwender SW (7)</td>
<td>Nicht verkehrstechnische Anwenderprogramme.</td>
</tr>
<tr>
<td>Versorgung (8)</td>
<td>Meldungen bei der Versorgung.</td>
</tr>
<tr>
<td>Uhr (9)</td>
<td>Meldungen bezogen auf Zeitfehler.</td>
</tr>
</tbody>
</table>
4.2.8.2 Definition von Meldungsteilen

Meldungen werden als Liste von MELDUNGSTEILen, die erweiterte STRUCTDOMAINs sind, definiert. Der Meldungsteil ist als IdData codiert. Über die darin festgelegten MEMBER und OTYPE Werte ist ein Meldungsteil eindeutig identifiziert. Die Parameter des Meldungsteils werden als Komponenten (DECL Einträge) abgebildet.


Die Zusatzinformationen zum Meldungstyp gibt der Hersteller in den entsprechenden Feldern an:

**Formatstring:** Zu jedem MELDUNGSTEIL kann es einen Formatstring geben, der die Meldung kurz charakterisiert. Es ist die Verantwortung der Zentrale, diesen Formatstring auszuwerten.

**MeldungsKategorie:** Durch die Kategorie wird der Fehlerort oder Absender eingeschränkt.

**MeldungsDegree:** Grad der Einschränkung der Verkehrsregelungsfunktion des Kreuzungsgerätes

4.2.8.3 Aufbau eines Meldungsteils

Alle Meldungsteile müssen von der MSGPART MELDUNGSTEIL abgeleitet werden. Die MSGPART ist nur eine spezielle STRUCTDOMAIN, bei der drei ClassAttributes vordefiniert sind: CATEGORY, DEGREE und FORMAT. CATEGORY enthält die Meldungskategorie als Zahl, DEGREE den MeldungsDegree als Zahl und FORMAT den Formatstring.

Es stellt sich natürlich die Frage, warum ein Meldungsteil überhaupt abgeleitet wird, da ja eine Meldung ohnehin schon in Meldungsteile untergliedert ist. Die Ableitung soll auch nicht dafür verwendet werden, semantisch ähnliche Meldungsteile voneinander abzuleiten, sondern soll nur tippfaulen Leuten ersparen, die Parameter immer wieder vollständig neu hinzuschreiben, wenn semantisch unterschiedliche Meldungen deklariert werden sollen.

Semantisch bedeutet die Ableitung: Die Basismeldung wird nicht zur Zentrale übertragen! Eine Ableitung schließt keine semantische Ableitung mit ein. Wenn also eine Basismeldung X existiert und eine Meldung Y von X abgeleitet wird, dann bedeutet das Übertragen von Y nicht, dass auch X gemeldet wurde. Stattdessen müssen neue Meldungsteile definiert werden.

OCIT-Outstations definiert eine Reihe von Meldungsteilen, die weiter unten aufgeführt sind.

4.2.8.4 Formattexte für Meldungsteile

Sinn des Formattextes ist, dass eine Zentrale herstellerspezifische Meldungen eines anderen Herstellers automatisch mit Hilfe der XML-Definition des betreffenden Herstellers/Gerätes ausgeben kann. Im Formattext wird lediglich ein kurzer charakteristischer Text für die Meldung hinterlegt. Wenn die Zentrale nicht über die XML-Definitionen verfügt, gibt sie diesen Text gefolgt von den Parametern in einfacher Form aus.

Beispiel:

Formattext: Detektorstörung, Name Parameter 1 DetektorNr, Parameter 2 DetektorName.

Ausgabetext: „Detektorstörung DetektorNr=7 DetektorName=Nebenrichtung“

Es gibt Meldungen welche ein Objekt referenzieren (z.B. Signalgruppe). Falls im Meldungstext der projektspezifisch versorgte Name ausgegeben werden soll, ist dieser in der Meldung als Stringparameter zu übergeben.

4.2.9 Welche Archive existieren?

Folgende Einträge sind in OCIT-Outstations fest vorgegeben:

In jedem Feldgerät:

<table>
<thead>
<tr>
<th>0</th>
<th>Das Betriebszustandsarchiv für die Speicherung des Betriebszustands</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Das Standard-Meldearchiv für allgemeine Störungsmeldungen wie Fehler (Störungsarchiv)</td>
</tr>
<tr>
<td>2</td>
<td>Das Syslog-Archiv für projektspezifische Informationen (allgemeines Archiv)</td>
</tr>
</tbody>
</table>

Optional:

| 3 | Ein Archiv für Service-Systemzugang |

4.2.10 Verhalten bei Stromausfall

Für alle Listen gibt das Listenattribut CurrentPersistenz das Verhalten bei Netzausfall an. Siehe Liste: SetSize().

Das Betriebszustandsarchiv und das Standard-Meldearchiv bleiben erhalten (Listenauftragsstruktur und Ringpufferinhalt CurrentPersistenz=Alles=2).

4.2.11 Übertragungsformat von Archivdaten (Format der Meldung)

Jeder Meldungsteil besteht aus einer Member/OType-Kennung, die die Meldung kennzeichnet. Auf die Member/OType-Kennung folgen die Länge des Parameterblocks und die Parameter, die zu der Meldung gehören.

4.2.12 Elementbeschreibungen Meldungsarchiv


Hauptmeldungsteile OCIT-Outstations (Member == 0):

(MeldungsDegree I: Information, W: Warnung, F: Fehler, S: Schwerer Fehler)

<table>
<thead>
<tr>
<th>OType</th>
<th>Kurzname</th>
<th>MeldungsDegree</th>
<th>Beschreibung</th>
</tr>
</thead>
<tbody>
<tr>
<td>60000</td>
<td>Störung aufgeho-</td>
<td>I</td>
<td>Diese Meldung kommt, wenn eine Störung vorgelegen hat und aufgehoben wurde.</td>
</tr>
<tr>
<td></td>
<td>ben</td>
<td></td>
<td>Sehr häufig durch zusätzliche Meldungsteile genauer spezifiziert. Dieser</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Hauptmeldungsteil wird nur verwendet für Meldungen für die keine spezielle</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Aufhebungsmeldung definiert ist.</td>
</tr>
<tr>
<td>60001</td>
<td>Netz aus</td>
<td>S</td>
<td>Zeigt an, zu welchem Zeitpunkt das Netz abgeschaltet wurde.</td>
</tr>
<tr>
<td>60002</td>
<td>Netz ein</td>
<td>I</td>
<td>Zeigt „Netzwiederkehr“ an.</td>
</tr>
<tr>
<td>60003</td>
<td>Systemfehler</td>
<td>S</td>
<td>Systemfehler die starke Auswirkung auf Gerätefunktion haben.</td>
</tr>
<tr>
<td>60012</td>
<td>Kommunikations-</td>
<td>W</td>
<td>Wird eingetragen, wenn die Kommunikation mit der zentrale gestört ist.</td>
</tr>
<tr>
<td></td>
<td>störung</td>
<td></td>
<td></td>
</tr>
<tr>
<td>60013</td>
<td>Kommunikation</td>
<td>I</td>
<td>Wird eingetragen, wenn die Kommunikation wieder aktiv ist.</td>
</tr>
<tr>
<td></td>
<td>ok</td>
<td></td>
<td></td>
</tr>
<tr>
<td>60016</td>
<td>Uhr gestört</td>
<td>W</td>
<td>Wird eingetragen, wenn die Uhr gestört ist. In diesem Fall ist es eine</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Holschuld des Gerätes, die Uhrzeit von der Zentrale zu be-</td>
</tr>
<tr>
<td>OType</td>
<td>Kurzname</td>
<td>MeldungsDegree</td>
<td>Beschreibung</td>
</tr>
<tr>
<td>-------</td>
<td>----------</td>
<td>----------------</td>
<td>--------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>schaffen, so dass die Synchronität gewahrt bleibt.</td>
</tr>
<tr>
<td>60017</td>
<td>Uhr ok</td>
<td>I</td>
<td>Die Uhr ist wieder in Ordnung.</td>
</tr>
<tr>
<td>60018</td>
<td>Wartung ein</td>
<td>I</td>
<td>Die Zentrale wird informiert, dass das Gerät gewartet wird.</td>
</tr>
<tr>
<td>60019</td>
<td>Wartung aus</td>
<td>I</td>
<td>Die Zentrale wird informiert, dass die Wartung beendet ist.</td>
</tr>
<tr>
<td>60020</td>
<td>Tür auf</td>
<td>W</td>
<td>Der Türschließkontakt meldet: Die Tür des Gerätes ist offen. (Die Meldung wird nur dann verwendet, wenn projektspezifisch der Türschließkontakt vorhanden ist)</td>
</tr>
<tr>
<td>60021</td>
<td>Tür zu</td>
<td>I</td>
<td>Der Türschließkontakt meldet: Die Tür des Gerätes ist geschlossen. (Die Meldung wird nur dann verwendet, wenn projektspezifisch der Türschließkontakt vorhanden ist)</td>
</tr>
<tr>
<td>60026</td>
<td>Zeitsprung</td>
<td>F</td>
<td>Meldet das Gerät, wenn es die Uhrzeit sprunghaft korrigiert. Der Zeitstempel der Meldung hat die neue Zeit. Parameter: Zeitdifferenz = Tnew – Talt als Sekunden (SLONG). Zeitquelle { Quartz, Zentrale, DCF, GPS }</td>
</tr>
<tr>
<td>60028</td>
<td>Suspend</td>
<td>I</td>
<td>Meldet, wenn eine Liste per Suspend angehalten wird. Parameter: Listennummer.</td>
</tr>
<tr>
<td>60029</td>
<td>Unsuspend</td>
<td>I</td>
<td>Meldet, wenn eine Liste per Suspend angehalten wurde und per Unsuspend wieder gestartet wird. Parameter: Listennummer.</td>
</tr>
<tr>
<td>60031</td>
<td>StopAuftrag</td>
<td>I</td>
<td>Meldet, dass ein Auftrag gestoppt wurde. Parameter: Listennummer, Auftragsnummer.</td>
</tr>
<tr>
<td>60032</td>
<td>ResetListe</td>
<td>I</td>
<td>Meldet, dass eine Liste RESETted wurde. Parameter: Listennummer.</td>
</tr>
<tr>
<td>60033</td>
<td>SyslogI</td>
<td>I</td>
<td>System Meldung Information. Parameter ist ein String.</td>
</tr>
<tr>
<td>60034</td>
<td>SyslogW</td>
<td>W</td>
<td>System Warnung.</td>
</tr>
<tr>
<td>60035</td>
<td>SyslogF</td>
<td>F</td>
<td>System Fehler.</td>
</tr>
<tr>
<td>OType</td>
<td>Kurzname</td>
<td>MeldungsDegree</td>
<td>Beschreibung</td>
</tr>
<tr>
<td>-------</td>
<td>----------------------</td>
<td>----------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>60036</td>
<td>SyslogSF</td>
<td>S</td>
<td>Schwerer Systemfehler</td>
</tr>
<tr>
<td>60039</td>
<td>TuerAufGeraete-</td>
<td>W</td>
<td>Optional ²: Geräteteil wurde geöffnet. Meldung wird als Zusatzmeldungsteil der Meldung TürAuf protokolliert.</td>
</tr>
<tr>
<td></td>
<td>teil</td>
<td></td>
<td></td>
</tr>
<tr>
<td>60044</td>
<td>TürZuBedienteil</td>
<td>I</td>
<td>Optional: Bedienteil wurde geschlossen. Meldung wird als Zusatzmeldungsteil der Meldung TürZu protokolliert.</td>
</tr>
<tr>
<td>60101</td>
<td>Sondermeldung 1-8</td>
<td>I</td>
<td>Projektspezifische Sondermeldung Nummer 1 bis 8</td>
</tr>
<tr>
<td>60109</td>
<td>Sondermeldung 9-16</td>
<td>W</td>
<td>Projektspezifische Sondermeldung Nummer 9 bis 16</td>
</tr>
<tr>
<td>60117</td>
<td>Sondermeldung 17-24</td>
<td>F</td>
<td>Projektspezifische Sondermeldung Nummer 17 – 24</td>
</tr>
<tr>
<td>60125</td>
<td>Sondermeldung 25-32</td>
<td>S</td>
<td>Projektspezifische Sondermeldung Nummer 25 – 32</td>
</tr>
</tbody>
</table>

Bei den Meldungen Reset, Suspend und Unsuspend gibt es folgende Parameter:

<table>
<thead>
<tr>
<th>Listennummer (UBYTE)</th>
<th>Nummer der Liste, die suspended wurde</th>
</tr>
</thead>
</table>

Bei den Meldungen StartAuftrag und StopAuftrag gibt es folgende Parameter:

<table>
<thead>
<tr>
<th>Listennummer (UBYTE)</th>
<th>Nummer der Liste, deren Auftrag bearbeitet wird.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auftragsnummer (UBYTE)</td>
<td>Nummer des Auftrags, der gestartet bzw. gestoppt wurde.</td>
</tr>
</tbody>
</table>

² OType 60039 bis 60044: Die Option für die erweiterten „Tür auf-Meldungen“ kann Hardwareanpassungen der Feldgeräte nach sich ziehen.
5 Abläufe Meldung und Messwerte

5.1 Listen mit vordefinierten Aufträgen

Die folgenden Listen haben vordefinierte Aufträge, welche lediglich modifiziert werden können:

- Das Betriebszustandsarchiv für die Speicherung des Betriebszustands
- Das Standard-Meldearchiv für allgemeine Störungsmeldungen wie Fehler (Störungsarchiv)
- Das Syslog-Archiv

Bei Reset der Listen werden die vordefinierten, ursprünglichen Aufträge wiederhergestellt.

5.1.1 Ziel

Es existiert eine ’gestartete’ Liste, von der dynamische Daten abgeholt werden können.

5.1.2 Ablauf

- **Reset der Liste (400:107)**: Die Funktion Reset der Liste hat zwei Bedeutungen. Bei nicht aktivierten Listen legt sie intern die Liste an und bei bereits laufenden Listen beendet sie den aktuellen Lauf und setzt die Werte auf die Defaults zurück. Das Ergebnis ist in jedem Fall eine frisch angelegte Liste.


• **Konfigurieren des Auftrags:** Das Konfigurieren des Auftrags ist vom Typ des Auftrags abhängig. Messwerte bestehen in der Regel aus mehreren Auftragselementen. Diese Auftragselemente müssen ebenfalls angelegt werden.

• **Anlegen eines Auftragselements (4xx:120):** Das Anlegen des Auftragselements geschieht für alle Auftragstypen identisch über die Funktion 120. Die Funktion gibt die Nummer des Auftragselements, das hinzugefügt wurde, zurück und erlaubt dann die Konfiguration des Auftragselements.

• **Festlegen des Event-Ziels (400:109):** Für jede Liste, die Events auslösen soll, wird ein Event-Ziel angegeben. Dies ist im Normalfall die Zentrale, es kann aber auch ein anderes Ziel eingetragen werden. Alle Events der Liste werden dann an dieses Ziel gesendet. Wenn kein Ziel eingetragen ist, wird per Default die Zentrale verwendet. Das Festlegen löst in jedem Fall einen Event bei der Zentrale aus, so dass die Zentrale weiß, welche Listen in Gebrauch sind.

• **Start der Liste:** Die Liste wird mit dem Befehl (400:105) aktiviert.

**5.2 Veränderung von Listen**

Das Ziel bei einer Veränderung von Listen ist, dass die alten Daten im Ringpuffer ohne Veränderung im Ringpuffer verbleiben können und trotzdem korrekt interpretiert werden. Man muss davon ausgehen, dass ein Client nur einen Teilbereich von Daten aus der Liste ausliest. Daraus wurde folgende Designentscheidung hergeleitet: **Ein Auftrag kann nicht gelöscht werden, ohne die Liste zu stoppen und neu zu starten (wobei der Puffer komplett gelöscht wird). Es ist allerdings möglich, dass ein Auftrag deaktiviert wird, so dass sich das Datenvolumen verringert.**

Wenn eine Liste geändert werden soll, sieht das Design lediglich ein komplettes Löschen der Liste und einen neuen Aufbau der Struktur vor. Der wesentliche UseCase ist eine Redefinition der Liste. Anstelle des Auslesens der Liste und einer Umdefinition kann die Liste gleich komplett neu definiert werden, was Aufrufe sparen kann.


**5.3 Wechseln des Grades (der Wichtigkeit) einzelner Meldungen**

Bei Meldungsarchiven wird pro Wichtigkeit der Meldung (dem sog. MeldungsDegree) ein Auftrag verwendet. Per Default sind alle Meldungen mit einem MeldungsDegree versehen. Der MeldungsDegree einer Meldung entspricht dem MeldungsDegree des Hauptmeldungsteils. Alle anderen Meldungsteile der Meldung sind für die Bestimmung des MeldungsDegree irrelevant.

Nun kann es projektspezifisch nötig sein, dass der MeldungsDegree gewechselt werden soll. Hierfür gibt es bei den Meldungsaufträgen die Funktionen IncludeMT und ExcludeMT. Mit IncludeMT wird eine Meldung einem Auftrag zugeordnet, mit ExcludeMT wird die Meldung aus einem Auftrag entfernt. Ein Meldungsteil, der in mehr als einem Auftrag vorhanden ist,
wird zweimal in den Ringpuffer geschrieben, ein Auftrag der in keiner Liste vorhanden ist, fällt unter den Tisch.

Um einen Grad zu wechseln, sind daher zwei Befehle nötig:

- Mit IncludeMT wird dem neuen Auftrag (und damit dem neuen Grad) die Meldung zugeordnet (genauer: Es wird die Kennung des Haupt-Meldungsteils der Meldung zugeordnet).
- Mit ExcludeMT wird dem alten Auftrag (und damit dem alten Grad) die Meldung entzogen.

Es ist notwendig, erst die Meldung neu zuzuweisen und dann zu löschen, damit eine Meldung, die während des Vorgangs auftritt, nicht verloren geht.

Wenn ein Element mit ExcludeMT aus einem Auftrag entfernt wurde, kann es mit IncludeMT wieder in den Auftrag aufgenommen werden. IncludeMT führt dabei selbständig das Löschen aus der Exclude-Liste durch.

5.4 Abholen von Daten

Meldungen und Messwerte werden auf die gleiche Art abgeholt. In keinem Fall werden Daten ungefragt vom Feldgerät in die Zentrale übertragen. Die Events melden nur die Abholbedürftigkeit an. Es gibt zwei unterschiedliche Anforderungen an das Abholen von Daten:

- Kontinuierliches Abholen der Daten. Beispiel: Die Zentrale holt in jedem Fall alle Betriebsmeldungen aus dem Gerät ab. Es ist unerheblich, ob die Daten einzeln oder geblockt abgeholt werden; sie werden in jedem Fall kontinuierlich alle aus dem Gerät abgeholt (Normalfall bei der Kommunikation Zentrale – Feldgerät).
- Spontanes Abholen von Teilen des Ringpuffers tritt auf, wenn z.B. Wartungsterminals Zugriff auf die Meldungen oder die Messwerte des Ringpuffers haben möchten (Normalfall Wartungsterminal – Feldgerät).


5.4.1 Kontinuierliches Abholen von Daten

Kontinuierliches Abholen von Daten wird nur von einem Gerät, im Normalfall der Zentrale, durchgeführt. Für das kontinuierliche Abholen wird die RIPID des letzten bereits abgeholten Elements benötigt. Sie wird der Funktion GetSFSince (bzw. GetSFSinceWithEvent, s.u.) als Parameter übergeben. Die Funktion gibt folgende Werte zurück:

- Eine Liste von Sekundenframes [Sekundenframe1..SekundenframeN]
- Die RIPID des ‚Sekundenframe0‘, der im Ringpuffer ‚vor‘ dem Sekundenframe1 steht.
• Die RIPID des SekundenframeN.

Wenn keine Elemente verlorengegangen sind, ist die zurückgegebene RIPID des Sekundenframe0 gleich der übergebenen RIPID. Andernfalls gingen Elemente verloren.

Es ist möglich, dass mit einem Call nicht alle neuen Elemente aus dem Ringpuffer ausgelesen werden. In diesem Fall sind noch Elemente im Puffer und es wird folgerichtig als RetCode SF_FOLLOW zurückgeliefert. Sind alle Elemente aus dem Puffer ausgelesen, wird in RetCode SF_NOFOLLOW zurückgemeldet.

Nur für das kontinuierliche Abholen ist die Funktion GetSFSinceWithEvent (400:103) gedacht. Bei dieser Funktion werden nicht nur Daten auf die gleiche Art wie bei GetSFSince abgeholt, sondern zusätzlich ein Event aktiviert, der auslöst, sobald die Liste (beginnend mit dem letzten zurückgegebenen Element) einen bestimmten Füllgrad überschreitet. Es wird also die Startposition für den EventOnFull Mechanismus markiert.

5.4.2 Spontanes Abholen von Teilen des Ringpuffers


• Das Wartungsterminal ruft die Funktion GetSFSince auf, aber mit ZStart-1 und einer garantiert ungültigen Position, dem Nullvalue.

• GetSFSince gibt eine Reihe von Sekundenframes zurück. Da nur die Anzahl der Sekundenframes übergeben wird, kann entweder das letzte Element im Zeitpunkt <= ZStop sein oder > als ZStop

• Solange die Zeit des letzte Element <= ZStop ist, werden mit Hilfe der RIPID des letzten zurückgegebenen Sekundenframes weitere Sekundenframes mit GetSFSince aus dem Ringpuffer geholt.


5.5 Feststellen, ob die Liste von außen (Systemzugang etc.) oder während einer Systemstörung geändert wurde

Die Zentrale (bzw. das Wartungsterminal) muss sich für jede Liste die Listenversion merken. Bei jeder Änderung der Liste, d.h. bei

• Änderung eines Auftrags,

• Änderung eines Auftragselements,
• Hinzufügen eines Auftragsbauteils und

• Hinzufügen eines Auftrags

wird die Listenversion geändert. Die Listenversion wird stets hochgezählt, also auch nach Reset. Wenn nach einem Ausfall der Versorgungsspannung die kompletten Listeninformationen gelöscht sein sollten, wird die Version mit 0 initialisiert.

Achtung: Beim reinen Starten und Stoppen von Aufträgen bleibt die Listenversion gleich.

Die Änderung der Listenversion erfolgt erst nach dem erneuten Starten des gestoppten Auftrags.

5.6 Ändern eines Auftrags während des Laufes

Anmerkung: Die vordefinierten Aufträge können nicht geändert werden.

Um einen Auftrag während des Laufes zu ändern, sind folgende Schritte nötig:

• Der Auftrag wird von der Zentrale gestoppt (4xx:122).

• Der Auftrag wird im Gerät geändert. (Hier ändert sich die Listenversion noch nicht)

• Der Auftrag wird von der Zentrale wieder gestartet. Bei einer Änderung wird hier die Listenversion hochgezählt (4xx:121).

Sobald der Auftrag wieder gestartet ist, wird die Listenversion auf den neuen Wert gesetzt. Es ist nicht notwendig, dass ein dynamischer Wert dieses Auftrags eingetragen sein muss.

Das Feldgerät weist alle Änderungsaufträge zurück, die nicht von der EventDestination stammen, so dass die Zentrale nicht ‚befürchten’ muss, dass die Listen unvermittelt von Wartungsterminals geändert werden.

5.7 Getriggertes Abholen von Daten

Es gibt drei verschiedene Typen von Triggern, die von Listen im Feldgerät zur Zentrale geschickt werden:

• OnFull bei Erreichen des Füllgrads

• OnInvalidate bei der Änderung der EventDestination

• OnInsert beim Eintragen bestimmter dynamischer Werte von ‚aktivierten’ Aufträgen.

Es ist die Aufgabe der Zentrale, auf die Events angemessen zu reagieren. Die Events enthalten keine Sachdaten, sondern ‚lediglich’ den Absender.

Bei OnFull und bei OnInsert müssen i.d.R. die Sekundenframes aus der entsprechenden Liste abgeholt werden.
Bei OnInvalidate hängt die Reaktion stark vom inneren Zustand der Zentrale ab. Die Änderung der EventDestination tritt sinnvollerweise bei folgenden Ereignissen auf:

- Die Zentrale ist ‚ausgefallen’ und ihre Funktion wird von einer Ersatzzentrale übernommen.
- Ein anderes Gerät als die Zentrale hat sich eine eigene Liste aufgebaut und möchte diese benutzen.
- Die Zentrale delegiert die Behandlung bestimmter Listen an ein anderes Gerät.

5.8 Abholen eines Datensatzes sofort nach Auftreten

Es gibt hier zwei unterschiedliche Anforderungen, die beide abgedeckt sind: Wenn aus einer Liste nur bestimmte Aufträge sofort abgeholt werden sollen, z.B. bei Meldungen nur Fehler und schwere Fehler, werden die Aufträge aktiviert. In diesem Fall wird der Event ausgelöst, wenn der Auftrag einen Wert in den Ringpuffer schreibt.

Wenn aus einer Liste immer alle Daten abgeholt werden sollen, nachdem sie eingetragen wurden, kann auch der Füllgrad auf 0% gesetzt werden. In diesem Fall wird der Event OnFull nach jedem neu eingetragenen Datensatz ausgelöst.

5.9 Synchronisieren nach Übertragungsstörung

Wenn ein GetSFSince fehlschlägt, muss das GetSFSince einfach wiederholt werden.

Wenn ein GetSFSinceWithEvent fehlschlägt, muss entsprechend die Funktion GetSFSinceWithEvent wiederholt werden.