

## 1. Einleitung

Die Prüfschrittfunktion „Kommunikation über Netzwerk-Socket“ erlaubt es ComGage, Daten an einen Netzwerk-Server zu senden. Die zu sendenden Daten werden in mehreren Textdateien konfiguriert, die Platzhalter enthalten können. ComGage ersetzt diese Platzhalter im laufenden Prüfplan durch aktuelle Daten, wenn die Prüfschrittfunktion ausgeführt wird.

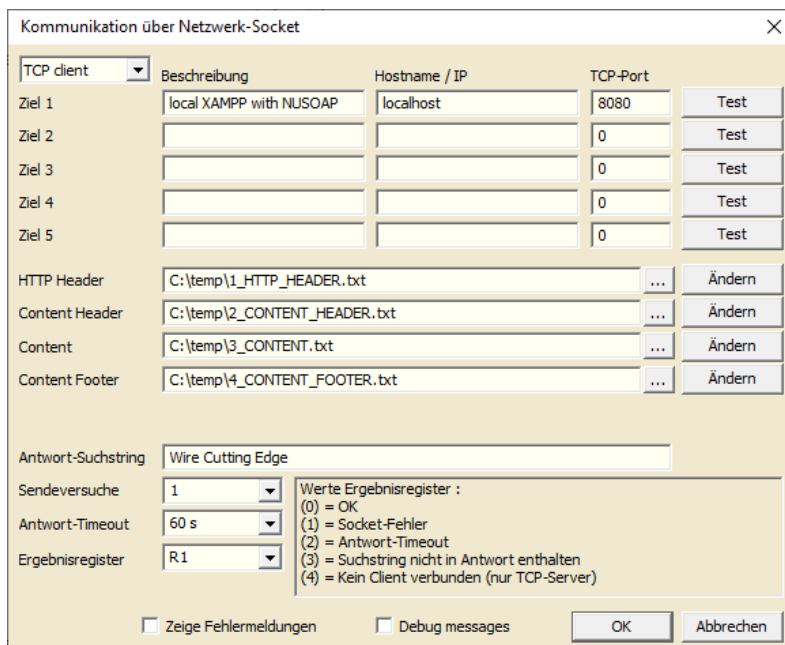
Wenn die Prüfschrittfunktion aufgerufen wird, kann der Benutzer einen der konfigurierten Zielsever aus einer Liste auswählen und einen Kommentar eingeben. Dieser Kommentar wird durch die Nutzung des zugehörigen Platzhalters ( siehe unten ) in den zu sendenden Daten ergänzt.

### Wichtige Hinweise:

- Diese Prüfschrittfunktion kann nur in einem Prüfauftrag ausgeführt werden !
- Um diese Prüfschrittfunktion nutzen zu können, wird die Lizenz für Modul 73 benötigt.
- ComGage V4.12 oder neuer wird benötigt, um diese Prüfschrittfunktion nutzen zu können.

## 2. Konfiguration

Die Prüfschrittfunktion muss in einem Prüfschritt eines ComGage Prüfplans ergänzt werden. Der Setup-Button der Funktion öffnet den folgenden Dialog :



Dieser Dialog erlaubt die Konfiguration der folgenden Parameter :

- **TCP client / TCP server**  
Hier kann gewählt werden, ob die SFct065 als TCP-Client ( zur Verbindung mit 1 bis 5 TCP-Servern ) oder als TCP-Server ( erlaubt im Messmode die Verbindung von TCP-Clients mit dem konfigurierten Port ) auftreten soll.
- **Ziel 1..5**  
Ermöglicht die Konfiguration von *Beschreibung*, *Hostname / IP* und *TCP-Port* für bis zu 5 verschiedene Ziele, falls *TCP client* ausgewählt wurde. Mit dem Button „Test“ kann die Verbindung zum jeweiligen Ziel mit den unten gewählten Einstellungen getestet werden. Die empfangene Antwort wird in einem Info-Fenster angezeigt.  
Falls *TCP server* ausgewählt wurde, kann nur der Port konfiguriert werden.
- **HTTP Header**  
Dateiname einer Textdatei, die den HTTP-Header ( z.B. eine HTTP post message ) enthält.<sup>1,2,3,4</sup>

<sup>1</sup> Platzhalter in dieser Datei werden durch aktuelle Werte ersetzt, wenn die Prüfschrittfunktion ausgeführt wird.

<sup>2</sup> Der Button ... öffnet einen Dialog, der die Auswahl einer Datei ermöglicht.

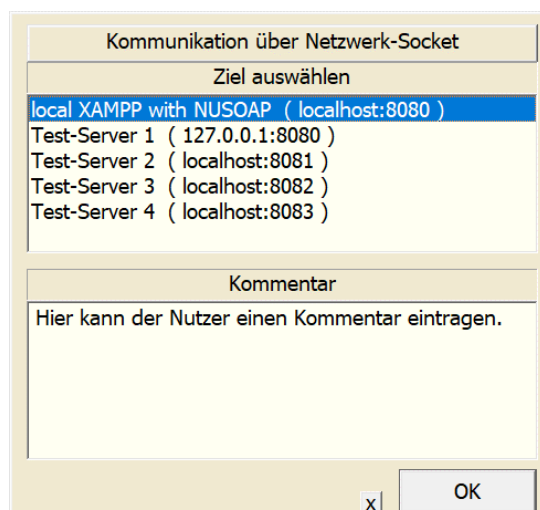
<sup>3</sup> Der Button Ändern öffnet die gewählte Datei zum Ändern im Standard-Editor des Systems.


<sup>4</sup> Die Länge des Dateinamens, inkl. Pfad, ist auf 200 Zeichen limitiert.

- **Content Header**  
Dateiname einer Textdatei mit dem Header des Nachrichteninhalts.<sup>1,2,3,4</sup>
- **Content**  
Dateiname einer Textdatei mit dem Nachrichteninhalt.<sup>1,2,3,4</sup>
- **Content Footer**  
Dateiname einer Textdatei mit dem Footer des Nachrichteninhalts.<sup>1,2,3,4</sup>  
**Hinweis :**  
Es müssen nicht alle 4 Dateien definiert werden. Falls nicht alle Dateinamen eingetragen wurden, werden nur die angegebenen Dateien gesendet.
- **Antwort-Suchstring**  
Falls hier ein String eingetragen wird, wird die empfangene Antwort nach diesem durchsucht.
- **Sendeversuche**  
Anzahl der Sendeversuche [1..255], bevor die Funktion einen Fehler meldet.
- **Antwort-Timeout**  
Die Zeit in Sekunden [1..255], in der das System eine Antwort erwartet. Wird in dieser Zeit keine Antwort empfangen, wertet das System dies als fehlgeschlagenen Sendeversuch.
- **Ergebnisregister**  
Enthält das Ergebnis des Funktionsaufrufs.  
(0) – Funktion erfolgreich ausgeführt.  
(1) – Socket-Fehler. Die Funktion konnte keine Verbindung zum gewählten Ziel aufbauen.  
(2) – Timeout, während auf eine Antwort vom Netzwerk-Server gewartet wurde.  
(3) – Die empfangene Antwort enthält den eingetragenen Suchstring nicht.  
(4) – Es sind keine Clients verbunden ( nur bei Einstellung *TCP server* ).
- **Zeige Fehlermeldungen**  
Ist diese Option aktiviert, werden Fehlermeldungen während der Ausführung der Funktion eingeblendet.
- **Debug messages**  
Ist diese Option aktiviert, schreibt die Funktion die Debug-Daten des letzten Aufrufs in die Datei `SFct065_debug.log` im konfigurierten ComGage-Datenverzeichnis für Prüfaufträge.

### 3. Dialogfenster bei Ausführung der Funktion

Falls die Funktion als TCP-Client konfiguriert wurde, wird bei der Ausführung im Prüfauftrag ein Dialog eingeblendet, der es dem Nutzer ermöglicht, eins der Ziele auszuwählen. Zusätzlich kann der Nutzer einen Kommentar eingeben. Mit dem OK - Button werden die vorkonfigurierten Informationen an den ausgewählten Netzwerk-Server geschickt :



Wird der Button  geklickt, werden keine Daten gesendet.

### 4. Platzhalter

Die folgende Tabelle enthält die Liste an zur Verfügung stehenden Platzhalter :

Platzhalter	Funktion
<\$HL>	Länge von Header + Inhalt + Footer des Nachrichteninhalts in Byte
<\$HC>	Kommentar ( wird vom Nutzer vor dem Senden der Daten eingegeben )
Platzhalter	Funktion
<\$ON>	Prüfauftrags-Nummer
<\$ON\$-B>	Prüfauftrags-Nummer ( Leerzeichen werden entfernt )
<\$ON\$L15>	Prüfauftrags-Nummer ( Länge=15 Bytes / fehlende Bytes=Leerzeichen )
<\$AR>	Artikel-Nummer
<\$AR\$-B>	Artikel-Nummer ( Leerzeichen werden entfernt )
<\$AR\$L15>	Artikel-Nummer ( Länge=15 Bytes / fehlende Bytes=Leerzeichen )
<\$AN>	Artikel-Bezeichnung
<\$AN\$-B>	Artikel-Bezeichnung ( Leerzeichen werden entfernt )
<\$AN\$L15>	Artikel-Bezeichnung ( Länge=15 Bytes / fehlende Bytes=Leerzeichen )
<\$DY>	Druck-Datum : Jahr ( 4-stellig )
<\$DM>	Druck-Datum : Monat ( 2-stellig )
<\$DD>	Druck-Datum : Tag ( 2-stellig )
<\$TH>	Druck-Uhrzeit :Stunde ( 2-stellig )
<\$TM>	Druck-Uhrzeit :Minute ( 2-stellig )
<\$TS>	Druck-Uhrzeit :Sekunde ( 2-stellig )

Merkmalsdaten :

Platzhalter	Funktion
<\$C1..C128\$RB>	Markiert den Anfang eines Messwert-Ausgabeblocks zur Ausgabe der vollständigen Messwert-Tabelle eines Merkmals. Der Inhalt des Ausgabeblocks wird für jeden gespeicherten Messwert des ausgewählten Merkmals wiederholt. Wenn zusätzliche Merkmale innerhalb des Blocks für die Ausgabe konfiguriert werden, wird für diese der gleiche Index ( = Messwertnummer ) verwendet.
<\$C1..C128\$RE>	Ende eines Messwert-Ausgabeblocks.
<\$C1..C128\$RI>	Index ( = Messwertnummer ) eines Messwertes innerhalb des Ausgabeblocks, beginnend bei 1.
Platzhalter	Funktion
<\$C1..128\$NA>	Merkmal 1..128 : Bezeichnung
<\$C1..128\$NA\$-B>	Merkmal 1..128 : Bezeichnung ( Leerzeichen werden entfernt )
<\$C1..128\$NA\$L15>	Merkmal 1..128 : Bezeichnung ( Länge=15 Bytes/fehlende Bytes=Leerzeichen )
<\$C1..128\$UN>	Merkmal 1..128 : Einheit
<\$C1..128\$UN\$-B>	Merkmal 1..128 : Einheit ( Leerzeichen werden entfernt )
<\$C1..128\$UN\$L15>	Merkmal 1..128 : Einheit ( Länge=15 Bytes / fehlende Bytes=Leerzeichen )
<\$C1..128\$NS>	Merkmal 1..128 : Nennmaß
<\$C1..128\$NS\$L12>	Merkmal 1..128 : Nennmaß ( Länge=12 Bytes / fehlende Bytes="0" )
<\$C1..128\$US>	Merkmal 1..128 : Obere Spezifikationsgrenze ( relativ zum Nennmaß )
<\$C1..128\$US\$L12>	Merkmal 1..128 : OSG ( Länge=12 Bytes / fehlende Bytes="0" )
<\$C1..128\$UC>	Merkmal 1..128 : Obere Eingriffsgrenze ( relativ zum Nennmaß )
<\$C1..128\$UC\$L12>	Merkmal 1..128 : OEG ( Länge=12 Bytes / fehlende Bytes="0" )
<\$C1..128\$LC>	Merkmal 1..128 : Untere Eingriffsgrenze ( relativ zum Nennmaß )
<\$C1..128\$LC\$L12>	Merkmal 1..128 : UEG ( Länge=12 Bytes / fehlende Bytes="0" )
<\$C1..128\$LS>	Merkmal 1..128 : Untere Spezifikationsgrenze ( relativ zum Nennmaß )
<\$C1..128\$LS\$L12>	Merkmal 1..128 : USG ( Länge=12 Bytes / fehlende Bytes="0" )
<\$C1..128\$UT>	Merkmal 1..128 : Obere Toleranzgrenze ( \$NS + \$US )
<\$C1..128\$UT\$L12>	Merkmal 1..128 : OT ( Länge=12 Bytes / fehlende Bytes="0" )
<\$C1..128\$LT>	Merkmal 1..128 : Untere Toleranzgrenze ( \$NS + \$LS )
<\$C1..128\$LT\$L12>	Merkmal 1..128 : UT ( Länge=12 Bytes / fehlende Bytes="0" )
<\$C1..128\$M1>	Merkmal 1..128 : Meisterwert 1
<\$C1..128\$M1\$L12>	Merkmal 1..128 : Meister 1 ( Länge=12 Bytes / fehlende Bytes="0" )
<\$C1..128\$M2>	Merkmal 1..128 : Meisterwert 2
<\$C1..128\$M2\$L12>	Merkmal 1..128 : Meister 2 ( Länge=12 Bytes / fehlende Bytes="0" )
<\$C1..128\$CO\$L15>	Merkmal 1..128 : Kal.-Offset ( Länge=15 Bytes / fehlende Bytes="0" )
<\$C1..128\$CF\$L15>	Merkmal 1..128 : Kal.-Faktor ( Länge=15 Bytes / fehlende Bytes="0" )
<\$C1..128\$NO>	Merkmal 1..128 : Bemerkung
<\$C1..128\$NO\$-B>	Merkmal 1..128 : Bemerkung ( Leerzeichen werden entfernt )
<\$C1..128\$NO\$L15>	Merkmal 1..128 : Bemerkung ( Länge=15 Bytes / fehlende Bytes=Leerzeichen )

Letzter Messwert aus Datei :

Platzhalter	Funktion
<\$C1..128\$MV\$5>	Merkmal 1..128 : Messwert \$5 : Anzahl der Nachkommastellen Ohne Angabe = 6 Nachkommastellen \$0 = ohne Nachkommastellen \$1 ... 5 = Anzahl Nachkommastellen gemäß Ziffer
<\$C1..128\$MV\$5\$L12>	Merkmal 1..128 : Messwert ( Länge=12 Bytes / fehlende Bytes="0" )
<\$C1..128\$RS\$xy\$z>	Merkmal 1..128 : Toleranz-Ergebnis des letzten Messwerts x : Export-Text bei Wert innerhalb Eingriffsgrenzen bzw. innerhalb Toleranzgrenzen bei deaktivierten Eingriffsgrenzen. y : Export-Text bei Wert außerhalb der Eingriffsgrenzen, aber innerhalb der Toleranzgrenzen z : Export-Text bei Wert außerhalb Toleranzgrenzen
<\$C1..128\$DY>	Merkmal 1..128 : Datum des Messwertes : Jahr ( 4 Ziffern )
<\$C1..128\$DM>	Merkmal 1..128 : Datum des Messwertes : Monat ( 2 Ziffern )
<\$C1..128\$DD>	Merkmal 1..128 : Datum des Messwertes : Tag ( 2 Ziffern )
<\$C1..128\$TH>	Merkmal 1..128 : Zeit des Messwertes : Stunde ( 2 Ziffern )
<\$C1..128\$TM>	Merkmal 1..128 : Zeit des Messwertes : Minute ( 2 Ziffern )
<\$C1..128\$TS>	Merkmal 1..128 : Zeit des Messwertes : Sekunde ( 2 Ziffern )
<\$C1..128\$R1..30>	Merkmal 1..128 : Stammdatensatz des letzten Messwerts aus Datei ( Prüfer, Charge, ... )
<\$C1..128\$R1..30\$-B>	Merkmal 1..128 : Stammdatensatz ( Leerzeichen werden entfernt )
<\$C1..128\$R1..30\$L15>	Merkmal 1..128 : Stammdatensatz ( Länge=15 Bytes / fehlende Bytes=Leerzeichen )  <b>!!! Die Stammdaten-Nummern R1...30 finden Sie im Menü „Grundeinstellung / Stammdaten“ !!!</b> <b>!!! Die Ausgabe von Stammdaten in einem Messwert-Ausgabeblock ist nicht möglich !!!</b>

ASCII-codes :

Platzhalter	Funktion
<00> ... <ff>	ASCII-code ( als HEX-Zahl ) eines Zeichens

### Beispiel 1 : Senden einer SOAP-Nachricht zum Web-Service [www.ptsv2.com](http://www.ptsv2.com)

Konfiguration der SFct065 wie in Kapitel 2 gezeigt, mit Nutzung von IP-Adresse und Port des SOAP-Servers :

#### Inhalt der Datei HTTP header.txt ( ANSI-Textdatei )

```
POST /post.php?dir=example2 HTTP/1.0
User-Agent: NuSOAP/4.22
Host: www.ptsv2.com
Content-Type: text/xml; charset=UTF-8
Content-length: <$HL>
```

#### Inhalt der Datei Content header.txt ( ANSI-Textdatei )

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
```

#### Inhalt der Datei Content.txt ( ANSI-Textdatei )

```
<ns6803:getProd xmlns:ns6803="http://tempuri.org">
  <category xsi:type="xsd:string">books</category>
</ns6803:getProd>
```

#### Inhalt der Datei Content footer.txt ( ANSI-Textdatei )

```
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Die versandte Nachricht kann unter [www.ptsv2.com](http://www.ptsv2.com) geprüft werden.

Zum Senden einer CSV-Liste aller gespeicherten Messwerte von Merkmal 1 kann folgende Vorlage verwendet werden :

```
<ns4159:getProd xmlns:ns4159="http://tempuri.org">
  <category xsi:type="xsd:string">
    Kommentar: <$HC>
    index;value;offset;factor;Year;Month;Day;Hour;Minute;Second;Machine
    <$C1$RB><$C1$RI>;<$C1$MV>;<$C1$CO>;<$C1$CF>;<$C1$DY>;<$C1$DM>;<$C1$DD>;<$C1$
    TH>;<$C1$TM>;<$C1$TS>;<$C1$R5>
    <$C1$RE></category>
  </ns4159:getProd>
```

### Beispiel 2 : Abrufen der Datei /index.html vom Webserver localhost

Nutzen Sie folgenden Inhalt für die HTTP-Header Datei, lassen Sie alle anderen Dateien leer.

```
GET /index.html HTTP/1.1
Host: localhost
```

**Hinweis :** Die Datei muss mit 2 Zeilenumbrüchen beendet werden. ( siehe [rfc7230](#) für Details ).