Datafox GmbH • Dermbacher Straße 12-14 • D-36419 Geisa • www.datafox.de

Datenprotokoll
HTTP- und HTTPS-Kommunikation

Flexible Datenerfassung mit Methode

Version 1.14.1 / Firmware Release 04.03.20.11 und neuer

Stände / Änderungen:			
Version	**Datum**	**Bearbeiter**	**Beschreibung, betroffene Kapitel und Seitenangabe**
V 1.0 – 1.4	14.11.2016	Bernd Ottmann	Dokument erstellt. Datenprotokoll Level 0 und Level 1 beschrieben.
V 1.5	19.03.2017	Sven Meyer	Kapitel 2.2 eingefügt und Level 1 komplettiert.
V 1.6	27.04.2017	Sven Meyer	Kapitel 2.2 überarbeitet, Indizes durch Objekt-Struktur ersetzt
V 1.7	04.07.2017	Sven Meyer	Kapitel 3 überarbeitet Kapitel 2.2.3: Entities für Parameter im http request ergänzt
V 1.8	10.10.2017	Sven Meyer	Kapitel 2.2.3: Repräsentation von Komma durch Entity in strukturierten Nachrichten ergänzt Kapitel 2.4.4 zu Puffergrößen ergänzt Kapitel 3 entfernt – es gibt jetzt eine eigene Software-Versionsliste Anhänge A und B ergänzt
V 1.9	25.06.2019	Sven Meyer	Kapitel 1.7 ergänzt 2.2.3.2.7 bis 2.2.3.2.7 und 2.2.3.2.10 bis 2.2.3.2.18 ergänzt Anhänge A, B.5-B.7, C, D und E ergänzt
V 1.9.1	06.12.2019	Sven Meyer	Anhang B.2: Falscher Hinweis auf nicht implementierte Client-Zertifikate entfernt.
V 1.10	08.07.2020	Sven Meyer Michael Wicher	Abschnitt 2.2.1.2 hinsichtlich Direkt-Übermittlung von IFF erweitert. Service-Mode Key in den Kapiteln 2.1.2.2.1 und 2.2.3.2.1 ergänzt. Kapitel 2.2.3.2.20 bis 2.2.3.2.24 ergänzt, COM.HTTP_MODE[.].SEND_IFF eingeführt. Anhang B.5 enthält nun einen Abschnitt zur Nutzung von ECC anstelle RSA Kapitel 2.3.1 (Systemmeldungen) um den Bereich Fingerprint erweitert.
V 1.11	23.12.2020	Sven Meyer	Kapitel 1.6 zur Basic Authentication eingefügt Bit-Definitionen in df_ac2 (2.2.3.2.8) an Dokumentation der Kommunikations-Bibliothek angeglichen (ab 04.03.16) Anhang B.5.3 ergänzt (Zertifikat im Microsoft Chromium Edge Browser) Anhang C überarbeitet (Initialkonfiguration eines Terminals über http) Anhang D überarbeitet (Datafox Beispiel-Webserver) Anhang E.5 ergänzt (Referenz der http.flags)
V 1.12	13.01.2022	Sven Meyer Michael Wicher Sven Meyer	Kapitel 1.7: Http-Header User-Agent ergänzt Abschnitt 2.2.3.2.8 um weitere Masken-Bits erweitert. Bits werden jetzt ab 1 gezählt Abschnitt 2.2.3.2.16 df_send_file: Token <image> teilw. umgesetzt Abschnitt 2.3.1: Systemmeldungscode 1037 eingefügt. Kapitel zu Systemmeldungen (2.3) und Verschlüsselung (2.4) klarer vom Protokoll-Inhalt abgetrennt. Anhang B.5.4: Hinweis zur Sicherung des client.key ergänzt. Anhang E.5: Flags zu abs_path und absoluteURI geändert, Bits werden jetzt ab 1 gezählt
V 1.12.1	03.06.2022	Sven Meyer	Abschnitt 2.2.3.2.16 df_send_file: Token <finger2> ergänzt Anhang A.3.1.3.1 – CRC-Berechnungsalgorithmus ergänzt Anhang C.3 ergänzt Anhang E.6 ergänzt
V 1.13	30.11.2022	Sven Meyer	Abschnitt 2.2.3.2.25 zum Firmware-Update ergänzt Abschnitt 2.3.1: Viele Systemmeldungen ergänzt Abschnitt 2.3.2: Ergänzt Anhang D: Dokumentation des Test-Webservers aktualisiert Anhang E Troubleshooting in Anhang T Troubleshooting umbenannt Neuen Anhang E zum Firmware-Update eingeführt.
V 1.14	15.02.2023	Sven Meyer	Anhang B.6: Erzeugen eigener, abgeleiteter Zertifikate Anhang E, Abschnitt 2.3.1: Behandlung des CPU-Modus in µSVC integriert, Abschnitt 2.3.1: Systemmeldungen 3915, 3916, 4502 und 4503 ergänzt. Anhang T.7: Nutzung von virtuellen Hosts zu Realisierung von Endpoints mit unterschiedlichen Zertifikatsketten

V 1.14.1	03.05.2023	Sven Meyer	Abschnitt 2.3.1: Systemmeldungen 1039 ergänzt Abschnitt 2.2.3.1.1 um HTTP-Status-Codes 200-299 erweitert. Anhang A.1: HTTP-Implementierung für DFCReset ergänzt.

Inhalt

1. Einleitung

In diesem Dokument ist die Kommunikation zwischen den Geräten und einer Web-Applikation per HTTP(S)-Protokoll beschrieben.

Hinweis:
Die Kommunikation mittels HTTP ist in den aktuellen Firmwareständen bei Geräten mit einer Kommunikation über TCP / IP möglich. Somit wird ein Modul für LAN / WLAN oder GPRS benötigt

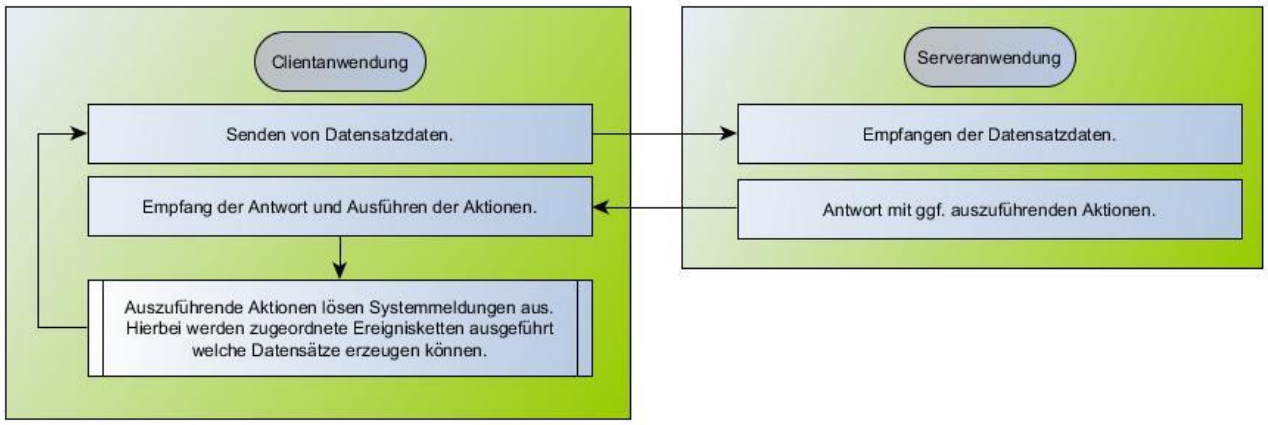
1.1. Erklärung der im Dokument verwendeten Begriffe.

Begriff	Erklärung
Webanwendung	Ist die Anwendung, welche die Anfragen der Geräte entgegennimmt und entsprechende Antworten zu den Geräten sendet.
Request	Anfrage, die von dem Gerät an die Webanwendung gesendet wird. Enthält z. B. die Daten zu einem Datensatz.
Response	Antwort auf einen Request, die von der Webanwendung an das Gerät gesendet wird. Enthält z. B. die Information, dass die Daten erfolgreich verarbeitet wurden oder Befehle, wie die Uhrzeit stellen, etc. auszuführen sind.
Client	In diesem Dokument ist die Gerätefirmware gemeint. Ggf. auch ein Web-Browser.
Server	In diesem Dokument ist hier der Web-Server gemeint, welcher die Webanwendung enthält.
Systemmeldung	Ein durch die Firmware erzeugtes Meldungsereignis. Dieses kann im Setup dazu verwendet werden, um über eine Ereigniskette einen Datensatz zu generieren. Speziell bei http können Systemmeldungen dazu verwendet werden, um die Ausführung gewünschter Aktionen (wie Listen aktualisieren) über einen Feedback-Datensatz zu quittieren.
Aktion	Unter einer Aktion ist ein auszuführender Vorgang oder einzelner Befehl zu verstehen. Z. B. ist das Übernehmen einer Uhrzeit, Setzen einer globalen Variablen oder Laden einer Listendatei eine jeweilige einzelne Aktion. Auszuführende Aktionen können durch die Webanwendung in der Response angegeben werden und führen im Client ggf. zu Systemmeldungen.

1.2. Formatierungen

Parameterangaben in Request oder Response werden kursiv dargestellt.

1.3. Grundlegendes Schema



- Ein Datensatz wird im Gerät aufgrund von Benutzerinteraktion oder Systemereignissen erzeugt.
- Der vorliegende Datensatz wird an den Server gesendet.
- In der Antwort des Servers können zusätzlich zur Bestätigung des Datensatzempfangs auszuführende Aktionen mit gesendet werden. Zum Beispiel eine Textnachricht soll angezeigt werden oder Listendaten sollen aktualisiert werden.
- Aufgrund der auszuführenden Aktionen werden Systemmeldungen ausgelöst, wodurch Datensätze im Gerät erzeugt werden, die dann wiederum an den Server gesendet werden können. Über diese sogenannten Feedback-Datensätze kann Ihre Webanwendung die Mitteilung über Zeitpunkt, Erfolg oder Misserfolg der auszuführenden Aktion erhalten.

1.4. Feedback – Datensätze

Im Gerätesetup können unterhalb der Signalverarbeitung „Systemmeldungen“ mit Ereignisketten verbunden werden. Über diese Ereignisketten können Sie Sich entsprechende Meldungsdatensätze auf die verschiedenen Ereignisse generieren. Welche Ereignisse existieren, wird bei den entsprechenden Funktionen weiter unten beschrieben.

! **Achtung:**
Bitte beachten Sie, dass keine Endlosschleifen entstehen. Wenn Sie z. B. auf einen gemeldeten Fehler, ohne diesen zu beheben, erneut dieselbe Aktion an das Gerät senden, kann es zu einer Endlosschleife kommen.

1.5. API - Level

Um Erweiterungen und Änderungen der Schnittstelle zu dokumentieren, wird mit einer „API – Level“ Angabe gearbeitet.

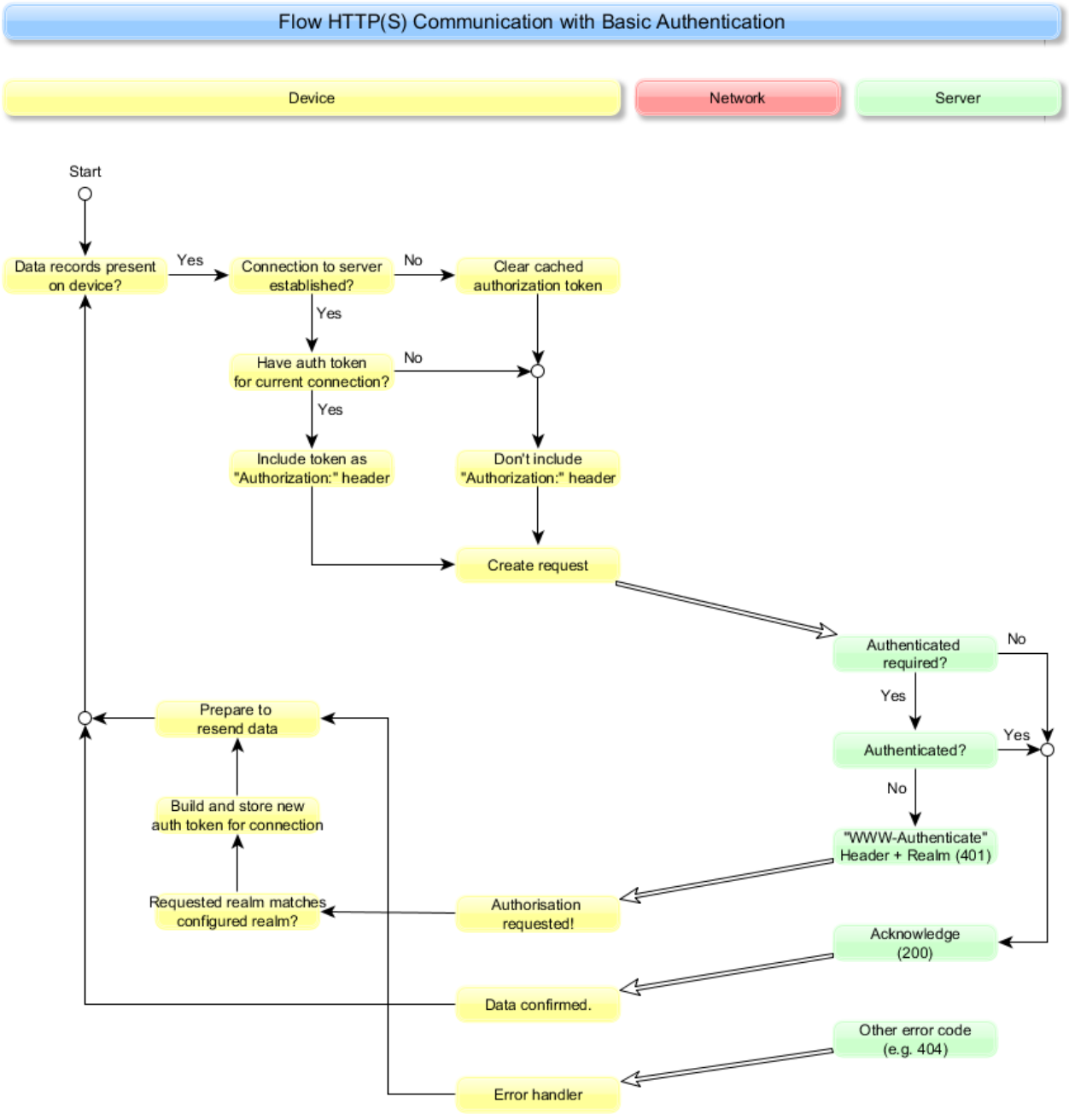
1.6. Basic Authentication

Basic Authentication ist eine Technologie im HTTP Kommunikationsablauf, bei der Server Anmeldeinformationen für einen Bereich (sog. Realm) anfordert. Diese Anmeldeinformationen werden – nach Aufforderung durch den Server – als Feld im HTTP-Header übermittelt.

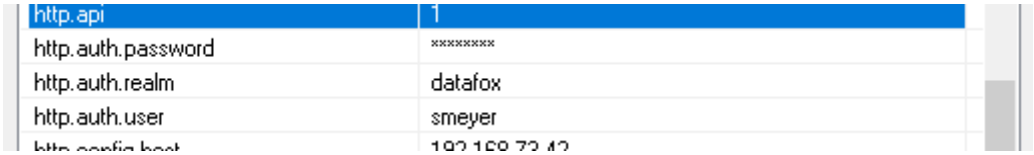
Die Technologie ist im Kontext von unverschlüsselter HTTP-Kommunikation als unsicher einzuordnen, da das Header-Feld nur chiffriert (aber nicht verschlüsselt) übermittelt wird und somit auf dem Transport abgegriffen werden kann.

Da die Technologie im HTTPS-Umfeld allerdings dennoch sicher genutzt werden kann, stellen Datafox Geräte dieses Protokoll ab Version 04.03.16 ebenfalls bereit.

Der vollständige Kommunikationsablauf zwischen einem Datafox-Gerät und Ihrem Web-Service stellt sich damit wie folgt dar:



Zur Konfiguration der Basic Authentication werden drei Systemvariablen eingesetzt, die z.B. über die Einstellungen zur Gerätekommunikation im Datafox Studio bearbeitet werden können:



Das Passwort wird während der Eingabe dargestellt.

Weiterführende Informationen finden Sie in RFC 7617: https://datatracker.ietf.org/doc/html/rfc7617.

1.7. HTTP-Kommunikation am Beispiel eines POST Requests

Das Gerät baut für zur Kommunikationsaufnahme einen HTTP-Request nachfolgendem Schema auf (hier am Beispiel eines http API Level 1 Requests dargestellt). Die veränderbaren Teile des Requests sind hier gelb unterlegt und im Folgenden erklärt:

```
POST https://extern.datafox.de:443/putdata HTTP/1.1
Host: extern.datafox.de:443
User-Agent: Datafox/04.03.18.04.http.10 11.3478
Accept-Charset: ISO 8859-1
Accept: application/x-www-form-urlencoded, text/html
Content-Length: 57
<weitere Header-Felder>

df_api=1&df_table=Alive&df_col_DT=2019-01-04T10%3A20%3A46
```

Hinweis:
Das Header-Feld „User-Agent“ wird mit folgendem Aufbau in Version 04.03.18.04 bereitgestellt:

`User-Agent: Datafox/`***`<Firmware inkl. Branch-Tag> <Dev-ID>.<SN>`***

Die fett kursiv geschriebenen Teile werden durch das Gerät beim Versand des http(s)-Requests eingetragen:

- Firmware inkl. Branch-Tag: Firmware-Version inkl. eines Tags für eine Feature- oder Bugfix-Version (z.B. „04.03.18.04.http.10“. Die Firmware-Version ist dabei „04.03.18.04“, das Branch-Tag „http.10“)
- Dev-ID: Interner Gerätetyp als Dezimalzahl (11 entspricht dem EVO 4.3)
- SN: Seriennummer als Dezimalzahl

Der Id des Gerätetypen bildet zusammen mit der Seriennummer eine eindeutige Kennung des Geräts.

Beispiele:

```
User-Agent: Datafox/04.03.18.04.http.10 11.3478
User-Agent: Datafox/04.03.18.05 11.3478
```

In diesem Request sind Felder enthalten, die über die Konfiguration des Geräts angepasst werden können:

- `POST https://extern.datafox.de:443/putdata HTTP/1.1`

  Sie können die Methode (GET oder POST) über den Beginn der Systemvariablen „com.http_mode[.].send“ vorgeben. Beginnt diese mit „POST“, so wird ein POST-Request erzeugt, sonst wird ein „GET“-Request erzeugt.

- `POST https://extern.datafox.de:443/putdata HTTP/1.1`

Die per TLS verschlüsselte Variante des http-Protokolls wird über die Systemvariable „com.http“ gesteuert. Wird hier auf HTTPS gewechselt, so erfolgt die Kommunikation per TLS abgesichert. Im Request wird dazu lediglich das „s“ bei der absoluteURI-Adressierung ergänzt, der Server-Port wird durch die Kommunikationseinstellung des Geräts vorgegeben.

- `POST https://extern.datafox.de:443/putdata HTTP/1.1`

  Gemäß RFC 2616 Sec. 5.2.1 gibt es abs_path- and absoluteURI-Adressierungen in der RequestLine. Hier dargestellt ist eine absoluteURI-Adressierung, eine abs_path-Adressierung wäre „`POST /putdata HTTP/1.1`“. Da absoluteURI nicht von allen Web-Servern korrekt verarbeitet wird – auch wenn für http/1.1 gefordert - können Sie die Adressierung über die Systemvariable „http.flags“ steuern (vgl. Anhang Troubleshooting)

- `POST https://extern.datafox.de:443/putdata HTTP/1.1`

  Der Typ des Requests wird über die Systemvariable “http.type” vorgegeben. Die unterstützten Typen „1.0“ und „1.1“ unterscheiden sich hinsichtlich des Verhaltens beim Schließen einer Verbindung – eine 1.0-Verbindung wird direkt nach jedem Request/Response-Paar geschlossen, die 1.1-Verbindung bleibt für nach einem Request/Response-Paar für einen weiteren Request zunächst geöffnet.

- `Host: extern.datafox.de:443`

  Der Host-Header eines HTTP-Requests kann – gem. RFC 2616 – den Port enthalten, auf den zugegriffen wird. Leider wird dieses nicht durch alle Web-Server unterstützt. Sie können die Übermittlung des Ports über die Systemvariable „http.flags“ steuern (vgl. Anhang Troubleshooting)

- Weitere Header-Felder können eingebunden werden, indem Sie die Datei „header.extensions“ auf das Gerät übertragen. Der Inhalt dieser Datei wird in jeden Request, den das Gerät sendet, eingebunden – sofern das dem Request-Typ entsprechende Flag in den „http.header_extension_flags“ gesetzt ist. Diese Variable enthält die aus der Summe der im Folgenden aufgeführten Werte, für die die Header-Flags übermittelt werden solle – als Dezimalzahl:
  - 1 = Datensatz-Requests
  - 2 = KVP-Requests
  - 4 = Listendaten-Download-Requests (Setup + Zutrittskontrolle)
  - 8 = IFF Download Requests
  - 32768 = übrige Requests

Achtung:
Bitte nutzen Sie die Header-Erweiterung nicht exzessiv. Die Gesamt-Header-Länge des Geräts ist begrenz, so dass Sie **nicht mehr als 500 Bytes** in den Header-Extensions ergänzen sollten!

!

Bitte beachten Sie, dass keine Konsistenz- oder Duplikats-Prüfung der so bereitgestellten Header-Felder durchgeführt wird. Sie dürfen auf diesem Weg nur Header-Felder ergänzen – es ist nicht vorgesehen, die Werte der oben dargestellten Felder zu verändern.

Bitte achten Sie ferner darauf, dass die „header.extensions“ mit CR und LF getrennt

sind, so wie dieses die http Spezifikation erfordert. Eine Umkodierung des Zeilentrenners durch die Firmware erfolgt nicht.

2. Beschreibung der jeweiligen API–Level

Aktuell werden zwei unterschiedliche Verfahren über HTTP zu kommunizieren unterstützt. Jedes dieser Verfahren hat einen entsprechenden Funktionsumfang.
Die unterschiedlichen Verfahren werden in Level 0 und Level 1 unterschieden, wobei das Verfahren zu Level 1 alle Möglichkeiten des Level 0 einschließt und somit die zu bevorzugende Variante darstellt.

2.1. Level 0

Achtung:
Das „Level 0“-Protokoll ermöglicht nur grundlegende Funktionen zur Gerätesteuerung. Es ist in der hier beschriebenen Form für Geräte der Generation Hardware 3 verfügbar.

Sollten Sie eine HTTP-Schnittstelle für aktuelle Geräte (Hardware 4) planen, setzen Sie bitte gleich das „Level 1“-Protokoll (siehe 2.2) ein.

In diesem Level werden die im Client generierten Datensätze an die Webanwendung gesendet. Über die Response der Webanwendung können Aktionen ausgeführt werden.

Klartext Anfrage
getdatagv.php?table=BB&bTYP=Manu&bLOG=Log&bDAT=2011-05-24_08:30:12&bPER=Per&checksum=2120
Klartext Antwort
status=ok&checksum=2120 (Prüfsumme der Anfage) (immer am Ende anzugeben ist: \r\n (carriage return line feed))

2.1.1. Request

Anfrage des Clients (Gerät) an den Server.

2.1.1.1. Methode: GET

Hinweis:
Wenn Sie einen festen Parameter z. B. eine Mandanten-Id benötigen, welche bei jedem Request mit gesendet wird, dann können Sie diesen in die URI der Systemvariablen MOBILE.HTTPSEND setzen.

Beispiel: GET /pfad/zum/script.php?**mandant=1234&**

Bitte achten Sie auf das abschließende „&“ und dass die Zeichenfolge der MOBILE.HTTPSEND Variablen eine Längenbegrenzung von 63 Zeichen hat.

Parametername	**Bedeutung**
table	Name der Datensatzbeschreibung, der die folgenden Datenfelder zugeordnet sind.
…	Zwischen *table* und *checksum* liegen die Felder der Datensatzbeschreibung. Die Parameternamen entsprechen den Feldnamen.

checksum	Prüfsumme über die Werte der einzelnen Datenfeldwerte. Siehe Kapitel „Parameter checksum“.

2.1.2. Response

Antwort des Servers an den Client (Gerät).

Content-Type: application/x-www-form-urlencoded; charset: iso-8859-1

2.1.2.1. Erforderliche Parameterangaben

<table>
<tr><th>Parametername</th><th>Bedeutung</th></tr>
<tr><td>status</td><td>Status der Verarbeitung.
<table>
<tr><td>ok</td><td>Es soll der nächste Datensatz geliefert werden. Die Prüfsumme muss in diesem Fall der gelieferten entsprechen (Prüfung optional) und muss in der Response auch so mit gesendet werden.
Es muss status=ok und die richtige checksum gesendet werden, damit auf den nächsten Datensatz gewechselt wird.</td></tr>
<tr><td>error</td><td>Der Datensatz soll erneut gesendet werden.</td></tr>
</table>
! Achtung:
Bitte stellen Sie sicher, dass keine Endlosschleife durch ständiges Senden von status=error entsteht. Der Client sendet den Datensatz so lange, bis er mit status=ok quittiert wird.</td></tr>
<tr><td>checksum</td><td>Prüfsumme über die Werte der einzelnen Datenfeldwerte. Siehe Kapitel „Parameter checksum“.</td></tr>
</table>

2.1.2.1.1. Parameter „checksum“

Bei der Prüfsumme handelt es sich um einen zusätzlichen Schutz, welcher die Datenkonsistenz der versendeten Feldwerte zwischen Client und Server sicherstellen soll. Die Daten selbst werden durch die Prüfsumme des TCP / IP sichergestellt. Es ist Ihnen überlassen, ob Sie die Prüfsumme validieren oder einfach den gelieferten Wert in der Antwort wieder zurücksenden.

Die Berechnung der Prüfsumme geschieht durch das Aufsummieren der einzelnen Zeichenwerte der Werte des GET-Requests. Die Keys gehen nicht in die Prüfsummenbildung ein!

Beispiel: …&feld1=4711&feld2=Kommt&… (Keys in blau, Values in rot)
4711 = 52 + 55 + 49 + 49
Kommt = 75 + 111 + 109 + 109 + 116
Durch Summenbildung der ANSI-Werte der einzelnen Zeichen ergibt sich eine Prüfsumme von 725.

2.1.2.2. Optionale Parameterangaben

Sie sind optional, können jedoch untereinander Abhängigkeiten aufweisen. Dieses wird durch eigene Tabellenblöcke dargestellt.

<table>
<tr><th>Parametername</th><th>Bedeutung</th></tr>
<tr><td>time</td><td>Datum und Uhrzeit, die übernommen werden sollen. Die Uhr wird jedoch erst bei einer Zeitdifferenz von mehr als +/- 10s übernommen.
Format: YYYY-MM-DD_hh:mm:ss
Beispiel time=2016-11-17_12:13:14</td></tr>
<tr><td>beep</td><td>Akustisches Signal ausgeben.
In der Tabelle wird ein ‚+’ dazu verwendet, einen langen Ton zu repräsentieren und ‚-, für einen kurzen Ton.
<table>
<tr><td>1</td><td>OK Signal</td></tr>
<tr><td>2</td><td>ERROR Signal</td></tr>
<tr><td>3</td><td>+</td></tr>
<tr><td>4</td><td>- +</td></tr>
<tr><td>5</td><td>- -</td></tr>
<tr><td>6</td><td>+ +</td></tr>
<tr><td>7</td><td>- - -</td></tr>
<tr><td>8</td><td>+ + +</td></tr>
<tr><td>9</td><td>- + -</td></tr>
<tr><td>10</td><td>+ - +</td></tr>
<tr><td>11</td><td>SMS Signal</td></tr>
</table></td></tr>
</table>

2.1.2.2.1. Service-Mode

<table>
<tr><th>Parametername</th><th>Bedeutung</th></tr>
<tr><td>service</td><td>Mit dem Wert 1 wird der Client dazu veranlasst nach dem Senden aller Datensätze in den Service-Mode zu wechseln.

Hinweis:
Die Verbindung wird per Standard durch die Webanwendung geschlossen. Erst wenn die Verbindung geschlossen wurde, kann der Client in den Service-Mode wechseln.

Um eine Verbindung zu beenden, können Sie im HTTP-Header der Response die Angabe „Connection: close“ mitgeben. Dadurch wird der Web-Server veranlasst, die Verbindung zu beenden.</td></tr>
<tr><td>host</td><td>Optional und nur angebbar, wenn service angegeben wurde.
Fehlt die Angabe wird auf die in der Systemvariablen com.http_mode[n].host hinterlegte zurückgegriffen.</td></tr>
</table>

port	Optional und nur angebbar, wenn *service* angegeben wurden. Fehlt die Angabe wird auf die in der Systemvariablen com.http_mode[n].port hinterlegte zurückgegriffen.
key	Optional und nur angebbar, wenn *service* angegeben wurden. Mit dem Parameter kann ausgewählt werden, ob die Service-Mode-Verbindung - unverschlüsselt aufgebaut werden soll (kein key-Parameter) - mit dem Schlüssel des ersten Aktive-Mode-Server verschlüsselt wird (key=key0) - mit dem Schlüssel des zweiten Aktive-Mode-Server verschlüsselt wird (key=key1) ☞ **Hinweis:** Der Parameter steht ab Firmware 04.03.14.09 zur Verfügung.

2.1.2.2.2. Globale Variable

Parametername	Bedeutung
setup.1	Setzen des Wertes einer globalen Variablen über ihren Index 1 – 8.
setup.id	Setzen des Wertes einer globalen Variablen über ihren Namen.

2.1.2.2.3. Ereigniskette

Parametername	Bedeutung
ek	Name einer Ereigniskette, welche ausgeführt werden soll.

2.1.2.2.4. Nachricht

Parametername	Bedeutung
message	Textnachricht, die am Display angezeigt werden soll. Einen Zeilenumbruch können Sie durch die Angabe von „\r“ (0x0d) erreichen. Beispiel: message=Dieses**\r**ist**\r**eine**\r**Nachricht. **! Achtung:** Die Nachricht wird nur angezeigt, wenn in dem eingesetzten Setup die Option „Server online“

	verwendet wird. Diese Option finden Sie auf der Seite zu den Grundeinstellungen.
delay	Gibt die Dauer in Sekunden an, wie lange die Nachricht angezeigt werden soll.
size	Gibt die Zeichensatzgröße und Art an. 0 Standardschrift 1 16 Pixel (7 Anzeigezeilen) 2 16 Pixel, feste Breite (7 Zeilen) 3 19 Pixel (6 Anzeigezeilen) 4 19 Pixel, feste Breite (6 Zeilen) 5 21 Pixel (5 Anzeigezeilen) 6 21 Pixel, feste Breite (5 Zeilen) Die in der Tabelle angegebenen Pixelwerte und Zeilen sind ca. Angaben und können je nach verwendetem Gerät variieren.

2.1.2.2.5. Onlinefunktion der Zutrittskontrolle

Parametername	Bedeutung
access, module	Der Wert der Zeichenfolge muss dem Format des Feldes "TM" der "Reader" Liste folgen. Er muss demnach immer 3 Ziffern umfassen.
master	Id für den RS485 Bus ZK, Beschreibt den ZK-Bus-Strang. RS485 Bus ID 1 RS485 Bus ID 2 usw. „master“ muss gemeinsam mit „module“ eingesetzt werden und ersetzt damit „access“
mask	1 / 0 Bei gesetztem Bit, wird der Buzzer angesprochen. 2 / 1 Bei gesetztem Bit, wird die grüne LED angesprochen. 4 / 2 Bei gesetztem Bit, wird die rote LED angesprochen. 8 / 3 Bei gesetztem Bit, wird das Relais 1 angesprochen. 16 / 4 Bei gesetztem Bit, wird das Relais 2 angesprochen. 32 / 5 Bei gesetztem Bit, wird das Relais 3 angesprochen. 64 / 6 Bei gesetztem Bit, wird das Relais 4 angesprochen. 128 / 7 Bei gesetztem Bit, wird das Relais 5 angesprochen. 256 / 8 Bei gesetztem Bit, wird das Relais 6 angesprochen. …. Unbenutzt. Bitte immer auf 0 setzen.
type	0 Aus 1 Ein 2 Wechsel (600ms an, 600ms aus)

	3	3 mal einschalten für 500ms
duration	Ist eine Zeitdauer und nur bei type = 1 gültig. Bedeutung: 0 = ständig ein, 1 - 40 = Sekunden ein.	

Achtung:
Die Reihenfolge „**access -> mask -> typ -> duration**" oder „**master -> module -> mask -> typ -> duration**" muss unbedingt eingehalten werden.

2.1.3. Verschlüsselung

Hinweis:
Wenn Sie HTTPS für die Gerätekommunikation einsetzen, bringt Ihnen der Einsatz der in diesem Kapitel beschriebenen Verschlüsselung keinen Sicherheitsgewinn.

Wir empfehlen daher, dieses Verfahren **bei Nutzung von HTTPS nicht einzusetzen**.

Die Datenfelder des Datensatzes können mittels eines Streamchiffre RC4 verschlüsselt werden. Dabei werden die (verschlüsselten) Feldinhalte dann in Hexadezimaldarstellung übertragen.

Parametername	Bedeutung
dfcb	Der Parameter gibt an, dass alle folgenden Felder bis einschließlich *dfce* verschlüsselte Feldinhalte haben. Der Wert von *dfcb* enthält den vierstelligen (1000-9999) Public-Key des anzuwendenden Passwortes für den Streamchiffre.
dfce	Der Parameter gibt an, dass alle folgenden Felder keine verschlüsselten Feldinhalte mehr haben. Wird der Wert korrekt entschlüsselt muss er mit dem Wert von *dfcb* übereinstimmen.

2.1.3.1. Veranschaulichung der GET-Anfrage

im Klartext (unverschlüsselt) und verschlüsselt:

Klartext Anfrage
getdatagv.php?table=BB&bTYP=Manu&bLOG=Log&bDAT=2011-05-24_08:30:12&bPER=Per&checksum=2120
Klartext Antwort
status=ok&checksum=2120(Prüfsumme)
Verschlüsselte Anfrage
getdatagv.php?dfcb=1000&table=e977&bTYP=14dce883&bLOG=4d7876&…&checksum=c01de865&dfce=019c1bd2
verschlüsselte Antwort
dfcb=1000&status=2b97&checksum=1726950d&…&setup_2=a449fd9c&setup_blue=a9375c8d0672&dfce=b99239f3

2.1.3.2. Erkennung einer Verschlüsselung

Um zu erkennen, ob die Datenfelder verschlüsselt versendet werden, wird der Anfang der Verschlüsselung mit ‚dfcb' (Datafox crypt begin) gekennzeichnet und mit ‚dfce' (Datafox crypt end)

das Ende gekennzeichnet. ‚dfcb' stellt das erste Feld im Request und ‚dfce' das letzte Feld im Request dar.

Der Wert des Feldes ‚dfcb' selbst wird im Klartext übertragen und ist der ‚public key'. Er ist eine Zufallszahl zwischen 1000 und 9999. Der Wert muss in Verbindung mit dem „Kommunikationspasswort" für die Ver- und Entschlüsselung herangezogen werden.

Die Chiffrierung der Daten erfolgt somit durch „private key + public key" als Passwortschlüssel.

In der Antwort muss das Feld ‚dfcb' 1:1 zurückgesendet werden. Damit wird sichergestellt, dass die Entschlüsselung erfolgreich war und die Antwort auch zur Anfrage passt.

Der Wert des Feldes ‚dfce' ist derselbe wie ‚dfcb', wird jedoch verschlüsselt übertragen. Beim Entschlüsseln kann somit sichergestellt werden, ob der verwendete Schlüssel korrekt ist. Der Wert von ‚dfce' muss daher nach dem Entschlüsseln gleich ‚dfcb' sein.

Gibt es Probleme bei der Entschlüsselung, muss als Antwort ‚dfc=error' gesendet werden. Zusätzlich sind die Felder ‚dfcb' und ‚dfce' mit Informationen zu bestücken.

Folgende Fehlerfälle sind durch das auswertende Script zu beachten:

‚dfcb' ist keine Zahl oder liegt außerhalb seiner Wertgrenze von 1000 – 9999

- Antwort: dfc=error&dfcb=range&dfce=unknown/missing
  - Range bedeutet Bereichsfehler, weil der Wert außerhalb seiner Wertgrenzen liegt.
  - Unknown bedeutet unbekannt, weil nicht ermittelt aber verfügbar
  - Missing bedeutet fehlt, es ist keine Angabe in der Anfrage vorhanden.

‚dfcb' ohne abschließendes ‚dfce'

- Antwort: dfc=error&dfcb=1000&dfce=missing

‚dfce' ist keine Zahl oder liegt außerhalb seiner Wertgrenze von 1000 – 9999

- Antwort: dfc=error&dfcb=1000&dfce=range

‚dfce' ohne beginnendes ‚dfcb'

- Antwort: dfc=error&dfcb=missing&dfce=unknown

‚dfce' ist ungleich ‚dfcb'

- Antwort: dfc=error&dfcb=1000&dfce=different
  - Different bedeutet ungleich, weil ‚dfce' nach Entschlüsselung ungleich ‚dfcb' ist.

2.1.3.3. Rückantwort des WEB-Servers

Die Feldinhalte der Anfrage werden nacheinander mit der RC4 Stromchiffre entschlüsselt. Die Feldinhalte der Rückantwort werden als Teil des Gesamtdatenstroms gesehen und werden im Anschluss an die Entschlüsselung mit der aktuellen Stellung der Stromchiffre wieder verschlüsselt. Einzige Ausnahme ist der erste Feldwert von ‚dfcb'. Dieser wird wie in der Anfrage 1:1, zurückgesendet.
Der Rückantwort muss als letztes verschlüsseltes Feld ‚dfce' angefügt sein. Der Wert von ‚dfce' (nach der Entschlüsselung) muss gleich dem Wert von ‚dfcb' sein.

Hinweis:
Bitte beachten Sie unbedingt, dass die Parameter von Request und Response in der Reihenfolge ent- bzw. verschlüsselt werden müssen, in der sie übertragen werden. Der RC4 Cipher generiert einen internen Zustand, so dass es nicht möglich ist, die Daten in beliebiger Reihenfolge zu verarbeiten.

2.2. Level 1

Achtung:

Für HTTP Level 1 wird ein Gerät mit Hardware V4 benötigt.

Allgemeine Änderungen zum Level 0:

- Alle speziellen Parameterangaben, im Request und Response, tragen den Vorsatz „df_“. Vorhandene Parameternamen wurden dem Schema gemäß angepasst.
- Bei Angaben von Datum und Uhrzeit wird als Trennzeichen nun ein ‚T’ anstatt des ‚_’ zwischen Datum und Uhrzeit verwendet. Dieses ist gemäß ISO-8601.
- Die Systemvariable com.http_mode[n].send kann wie folgt angegeben werden.
  - Beispiel für Level 0: „GET /pfad/zum/script.php?“
  - Beispiel für Level 1: „/pfad/zum/script.php“

2.2.1. Änderungen im Request und Response zu Level 0.

HTTP API Level 1 unterscheidet zwei Arten von Requests. Der Request, mit dem Datensätze übermittelt werden, ist vergleichbar zum Level 0 Request. In der Rückantwort auf den Request werden dann Steuer-Befehle vom Server zum Gerät übermittelt – diese sind in den Unterkapiteln des Abschnitts 2.2.3 zusammengestellt.
Den anderen Request-Typ stellt die Übermittlung von Binärdaten vom Gerät zum Web-Server dar. Hier werden Daten typischer Weise IFF-codiert an den Web-Server übermittelt, so dass eine URL auf dem Web-Server unterschiedliche Datenpakete erhält und unterscheiden kann.

2.2.1.1. Änderungen zum Level 0 Request

- Als erstes Feld wird immer *df_api=1* gesendet. Dieses Feld ist bei einer ggf. aktiven Verschlüsselung unverschlüsselt.
- Der Parameter *table* heißt nach dem Namenschema nun *df_table*.
- Alle Datenfelder beginnen nun nach dem Namenschema mit „*df_col_*“.
- In Datenfeldern vom Typ „Datum und Uhrzeit“ sind Datum und Uhrzeit mit einem ‚T’ anstatt ‚_’ getrennt.
- Der Parameter *checksum* entfällt komplett.
- Bei einer aktiven Verschlüsselung werden die Parameter *dfcb* und *dfce* nach dem neuen Namensschema als *df_cb* und *df_ce* gesendet.

2.2.1.2. Änderungen zum Level 0 Response

Es gibt zwei Modi eine Response in API-Level 1 zu senden. Sie können, so wie bisher, Instruktionen zusammenstellen, die das Gerät ausführen soll. Dann verändert sich die Response wie folgt:

- Der Parameter *status* und *checksum* entfällt komplett.
- Als erstes Feld muss immer *df_api=1* gesendet werden. Dieses Feld ist bei einer ggf. aktiven Verschlüsselung unverschlüsselt zu übertragen.
- Alle vorhandenen Parameter wurden mit dem Präfix „*df_*“ versehen und ggf. im Namen an die aktuell gebräuchlichen Begriffe angepasst.

- Bei einer aktiven Verschlüsselung werden die Parameter *dfc*, *dfcb* und *dfce* nach dem neuen Namensschema als *df_c*, *df_cb* und *df_ce* gesendet.
  Bitte beachten Sie ferner, dass dieses Verschlüsselungssystem recht schwach ist und HTTPS auf Hardware-IV Geräten verfügbar ist. Daher ist die Beschreibung in Abschnitt 2.4 ausgegliedert worden.
- Am Ende der Antwort **muss kein** Carriage Return/Line Feed mehr gesendet werden. Erfolgt dieses dennoch, so wird das letzte paar CR/LF entfernt. Werden CR/LF in der Nachricht benötigt, so sind diese zu kodieren:
  - Alle Zeichen außer Buchstaben, Ziffern, - (Minus), . (Punkt), _ (Unterstrich) und ~ (Tilde) sind als %xx zu kodieren, wobei xx der hexadezimale ASCII Code des zu kodierenden Zeichens ist [RFC 3986 Abschnitt 2.3 / 2.4]
- Dateien, die vom Gerät heruntergeladen werden sollen, können nur vom Webserver bereitgestellt werden, an den die Anfrage gesendet wurde. Es ist nicht möglich, hier durch das Gerät eine Anfrage zu einem anderen Webserver zu senden.
- Pfadangaben, mit denen ein Download ausgelöst wird, werden immer absolut auf dem Webserver betrachtet.

Alternative zum „df_api=1“-Response können Sie ein **IFF-Transfer-Datei** gemäß Anhang A.3: Aufbau einer Transferdatei senden. Genügt die übermittelte Datei formal den Anforderungen an den IFF-Aufbau, so wird der im Request übermittelte Datensatz als quittiert betrachtet. Das Gerät wertet direkt die IFF-Datei aus und übernimmt die darin kodierten Daten.

2.2.1.3. Übermittlung von Bildern und langen Barcodes (ab 04.03.18.04)

Für die Übertragung von Bildern (wie dieses bei der Unterschriften-Erfassung oder durch die Stichproben-Kamera erzeugt werden) und langen Barcodes, wird – sofern die Systemvariable COM.HTTP_MODE[.].SEND_IFF einen nicht-leeren Wert aufweist - eine IFF-Upload-Datei vom Typ Bild erzeugt (siehe auch Anhang A.3 Transfer-Datei, Typ 0xDF06) und an den Server übermittelt. Diese Übermittlung erfolgt einmalig sobald ein Bild/langer Barcode auf dem Gerät erzeugt wird. Besteht zu diesem Zeitpunkt keine Verbindung zum HTTP-Server, so wird das Bild nicht übermittelt – es kann später per df_send_file explizit angefordert werden.

Hinweis:
Mit Firmware Release 04.03.20.06 wurde ergänzt, dass direkt nach der Übermittlung eines Barcodes oder Bildes an den Server, ein Alive-Datensatz erzeugt wird. Wir haben im Labor (HTTPS über LAN) ermittelt, dass der Alive ~ 200 ms nach dem IFF-Upload beim Server eingeht. Die Laufzeit hängt natürlich von verschiedenen Faktoren ab, so dass diese im Kundeneinsatzszenario abweichen kann.

2.2.2. Request

Anfrage des Clients (Gerät) an den Server.

2.2.2.1. Methode: GET

Hinweis:
Wenn Sie einen festen Parameter z. B. eine Mandanten-Id benötigen, welche bei jedem Request mit gesendet wird, dann können Sie diesen in die URI der Systemvariablen com.http_mode[n].send setzen.

Beispiel: /pfad/zum/script.php**?mandant=1234**

Bitte achten Sie darauf, dass die Zeichenfolge eine Längenbegrenzung von 63 Zeichen hat.

Parametername	Bedeutung
df_table	Name der Datensatzbeschreibung zu welcher die folgenden Datenfelder zugeordnet sind.
df_record_state	Kennung, ob der Datensatz aktuell („online“) und erstmalig übertragen wird: 1 = Online 2 = Wiederholt online 3 = Offline 4 = Wiederholt offline Anmerkung: Ob und wie der `df_record_state` gesendet wird, kann über die Systemvariable `http.record_state` festgelegt werden. **Hinweis:** Die Kennung des Record-State ist nur für Datensätze der Bedienung des Geräts verfügbar. Systemmeldungen, Alive-Datensätze, Datensätze der Zutrittskontrolle, etc. enthalten diese Kennung nicht!
df_col_{Feldname}	Wert des Feldes der Datensatzbeschreibung. Der Name des Feldes wie er in der Datensatzbeschreibung angegeben ist, wird dem statischen Teil „*df_col_*“ angefügt. „*col*“ steht hierbei für „column“. Bei Feldern vom Typ „Datum Uhrzeit“ wird der Inhalt wie folgt dargestellt: Format: YYYY-MM-DDThh:mm:ss Beispiel time=2016-11-17T12:13:14

2.2.3. Response

Antwort des Servers an den Client (device).

Content-Type: application/x-www-form-urlencoded; charset: iso-8859-1

Mit der Antwort können Anweisungen aus der folgenden Übersichtstabelle an das Gerät übermittelt werden.

Parametername	Bedeutung
df_time=2016-11-17T12:13:14	Datum Uhrzeit am Gerät stellen.
df_beep=1 (1-11)	OK-Signal / Beep am Gerät erzeugen

Parametername	Bedeutung
df_service=1,www.datafox.de,10047	Verbindung zur Kommunikations-Bibliothek im Aktiv-Modus herstellen. Auch zum DatafoxStudioIV möglich. Angabe von IP/URL und Port möglich.
df_var=setup.1,wert	Wert einer GV im Setup ändern.
df_ek=name	Eine Aktion im Gerät auslösen. Starten eine Ereigniskette in der Signalverarbeitung.
df_msg=Dieses\rist\reine\rNachricht,5,1,0	Textnachricht auf das Display senden.
df_msg_icon=2	Stelle folgende Nachrichten auf dem Display mit dem Icon einer F-Kette (hier F2) dar.
df_backlight=0,5,255,255,0,192	Stellt ein Geräte-Backlight für eine gewisse Zeit auf eine bestimmt Farbe (RGBW) ein.
df_info_msg=Info\rNachricht,0	Stellt die Info-Nachricht des Geräts ein.
df_ac2=010,1,10,20,5	AC = access control. Aktionen in der Zutrittskontrolle auslösen.
df_custom_msg_ac2=010,1,1,0,Hallo%20Welt	Sendet eine Nachricht an einen Bus-Teilnehmer in einer Zutrittskontroll-Installation
df_ao_ac2=0,1234	Quittiert eine Aktion der Zutrittskontrolle mit Vorprüfung
df_trigger_ac2=1,011,6543210,0	Simuliert den Eingang einer Buchung von einem Leser am ZK-Controller
df_kvp=var,ID	Fordert einen Wert vom Gerät an. Dieser wird dann als Key-Value-Paar an den Server übermittelt.
df_set_relay=2,close,5	Setzt den Zustand eines nicht von der Zutrittskontrolle verwendeten Relais
df_toggle_relay=2,5	Ändert den Zustand eines nicht von der Zutrittskontrolle verwendeten Relais
df_load_file=/path/on/server	Veranlasst das Herunterladen einer Datei vom Server
df_send_file=/logs/,syslog,0	Veranlasst das Übermitteln einer Datei an den Server
df_remove_file=root:datafox.cert	Veranlasst das Löschen einer Datei durch das Gerät
df_remove_finger=1980,all	Löschen von Fingern aus dem Fingerprint-Sensor
df_setup_list=Personal,/Pfad/zur/liste.txt	Dem Gerät eine neue Liste z.B. Personal geben.
df_ac2_list=Identification,/pfad/zur/liste.txt	Dem Gerät eine neue Liste ZK-Liste geben.
df_table_count=list.PID	Liefert die Anzahl der Einträge in einer Liste
df_table_select=list.PID,/upload/form,Abteilung=Entwicklung,PID=5	Wählt einen oder mehrere Datensätze aus einer Liste zum Upload aus

Parametername	Bedeutung
df_table_append=list.PID,9999,,Besucher,	Fügt einen Datensatz an eine Liste an
df_table_update=list.PID,,,Abteilung=	Ändert Werte in einer Liste
df_table_delete=list.PID,Abteilung=Entwicklung	Löscht Zeilen aus einer Liste

Hinweis:
Wie beim URL-Encoding durch den Web-Browser können mit der URL Parameter (Instruktionen) gemäß der folgenden Aufstellungen an das Terminal übertragen werden. Die Parameter sind aus Name-Wert-Paaren zusammengesetzt. Zwischen Name und Wert findet sich ein ‚=', die Paare werden durch ‚&' getrennt.

Sollte ein Name oder Wert ein ‚=', ‚?', ‚&' oder Komma enthalten, so müssen diese durch die zugehörigen Entities ersetzt werden, also %3d (‚='), %3f (‚?'), %26 (‚&') oder %2C (Komma).

2.2.3.1. Erforderliche Parameterangaben

Parametername	Bedeutung
df_api	Verwendeter API-Level der Antwort. Nutzen Sie hier 1. **Achtung:** Bitte stellen Sie sicher, dass keine Endlosschleife durch ständiges Senden von fehlender oder falscher Angabe entsteht. Der Client sendet den Datensatz so lange, bis er mit *df_api=1* und dem HTTP-Result „200 OK" quittiert wird.

2.2.3.1.1. Quittieren von Datensätzen

Eine Quittierung des Datensatzes liegt vor, wenn in der Response die Parameterangabe *df_api=1* vorliegt und der HTTP-Result „200 OK" ist. Fehlt die Angabe von *df_api* meldet der HTTP-Server einen Status-Code verschieden von 200, wird der Datensatz im Gerät nicht quittiert und somit erneut gesendet.

Seit Firmware-Version 04.03.20.11 können Sie mit Status-Codes 201-299 ausdrücken, dass zwar der Datensatz nicht quittiert werden soll, allerdings die Instruktionen der Response (siehe folgende Abschnitte) ausgewertet und verarbeitet werden sollen.

Anstelle eines Protokoll-Responses „df_api=1&…" kann auch direkt eine gültige IFF-Datei im Sinne von Anhang A.3 übermittelt werden. Erkennt das Gerät diese, so wird der formale Aufbau der IFF-Datei geprüft. Ist dieser gegeben, so wird der Datensatz quittiert und der Inhalt der IFF-Datei ausgewertet / verarbeitet.

2.2.3.2. Optionale Parameterangaben (df_time und df_beep)

Sie sind optional, können jedoch untereinander Abhängigkeiten aufweisen. Dieses wird durch eigene Tabellenblöcke dargestellt.

<table>
<tr><th>Parametername</th><th>Bedeutung</th></tr>
<tr><td>df_time</td><td>Datum und Uhrzeit, die übernommen werden sollen. Die Uhr wird jedoch erst bei einer Zeitdifferenz von mehr als +/- 10s übernommen.
Format: YYYY-MM-DDThh:mm:ss
Beispiel time=2016-11-17T12:13:14</td></tr>
<tr><td>df_beep</td><td>Akustisches Signal ausgeben.
In der Tabelle wird ein ‚+’ dazu verwendet, einen langen Ton zu repräsentieren und ‚-, für einen kurzen Ton.
<table>
<tr><td>1</td><td>OK Signal</td></tr>
<tr><td>2</td><td>ERROR Signal</td></tr>
<tr><td>3</td><td>+</td></tr>
<tr><td>4</td><td>- +</td></tr>
<tr><td>5</td><td>- -</td></tr>
<tr><td>6</td><td>+ +</td></tr>
<tr><td>7</td><td>- - -</td></tr>
<tr><td>8</td><td>+ + +</td></tr>
<tr><td>9</td><td>- + -</td></tr>
<tr><td>10</td><td>+ - +</td></tr>
<tr><td>11</td><td>SMS Signal</td></tr>
</table></td></tr>
</table>

2.2.3.2.1. Service-Mode (df_service)

<table>
<tr><th>Parametername</th><th>Bedeutung</th></tr>
<tr><td>Flag</td><td>Mit dem Wert 1 wird der Client dazu veranlasst nach dem Senden aller Datensätze in den Service-Mode zu wechseln.

Mit dem Wert 2 startet der Client die Service-Mode-Verbindung auch wenn noch Datensätze auf dem Gerät vorhanden sind.

☞ Hinweis:
Die Verbindung wird per Standard durch die Webanwendung geschlossen. Erst wenn die Verbindung geschlossen wurde, kann der Client in den Service-Mode wechseln.

Um eine Verbindung zu beenden, können Sie im HTTP-Header der Response die Angabe „Connection: close“ mitgeben. Dadurch wird der Web-Server veranlasst, die Verbindung zu beenden.</td></tr>
</table>

Host	Fehlt die Angabe wird auf die in der Systemvariablen com.http_mode[n].host hinterlegte zurückgegriffen.
Port	Fehlt die Angabe wird auf die in der Systemvariablen com.http_mode[n].port hinterlegte zurückgegriffen.
Key	Mit dem Parameter kann ausgewählt werden, ob die Service-Mode-Verbindung - unverschlüsselt aufgebaut werden soll (kein oder leerer Parameter) - mit dem Schlüssel des ersten Aktive-Mode-Server verschlüsselt wird („key0“) - mit dem Schlüssel des zweiten Aktive-Mode-Server verschlüsselt wird („key1“) ☞ **Hinweis:** Der Parameter steht ab Firmware 04.03.14.09 zur Verfügung.

Das Aktivieren des Service-Modus erfolgt durch übermitteln des Parameters

```
df_service=<Flag>,<Host>,<Port>,<Key>
```

im Response des Servers, etwa

```
df_service=1,active-mode-server.my.net,8000
df_service=1,second-active-mode-server.my.net,8000,key1
```

2.2.3.2.2. Globale Variable, Setup- oder System-Variable setzen (df_var)

Parametername	Bedeutung
Name	Name einer globalen Variablen mit ihren Index 1 – 8. Beispiel: setup.1 Name einer globalen Variablen. Beispiel: setup.GlobVar1
Wert	Zu setzender Wert der Variablen

Zum Setzen einer globalen Variablen fügen Sie in der HTTP-Response den Parameter

```
df_var=<Name>,<Wert>
```

ein, also beispielsweise

```
df_var=setup.1,4711
```

2.2.3.2.3. Ereigniskette (df_ek)

Parametername	Bedeutung
Name	Name einer Ereigniskette, welche ausgeführt werden soll. ☞ **Hinweis:** Ereignisketten sind die Ketten, die im Rahmen der Signalverarbeitung genutzt werden können. Diese ermöglichen keine Benutzerinteraktion. Wenn Sie Nachrichten an den Terminal-User versenden wollen, wie etwa das Saldo als Antwort auf eine Buchung, nutzen Sie bitte df_msg.

```
df_ek=<Name>
```

2.2.3.2.4. Nachricht am Display anzeigen (df_msg)

Parametername	Bedeutung
Nachricht	Textnachricht, die am Display angezeigt werden soll. Einen Zeilenumbruch können Sie durch die Angabe von „\r“ erreichen. ! **Achtung:** Die Nachricht wird nur angezeigt, wenn in dem eingesetzten Setup die Option „Server online“ verwendet wird. Diese Option finden Sie auf der Seite zu den Grundeinstellungen. Ferner ist es erforderlich, dass „df_msg“ als **Antwort auf einen Online Datensatz der Bedienung** gesendet wird – eine Systemmeldung oder ein Alive-Datensatz sind **nie** Online-Datensätze.
Dauer	Gibt die Zeit in Sekunden an, wie lange die Nachricht angezeigt werden soll.
Beep	Akustisches Signal ausgeben. In der Tabelle wird ein ‚+’ dazu verwendet, einen langen Ton zu repräsentieren und ‚-, für einen kurzen Ton. 0 – Kein Signalton 1 – OK Signal 2 – ERROR Signal 3 – + 4 – - + 5 – - - 6 – + + 7 – - - - 8 – + + +
Font	Gibt die Zeichensatzgröße und Art an.

	<table><tr><td>0</td><td>Standardschrift</td></tr><tr><td>1</td><td>16 Pixel (7Anzeigezeilen)</td></tr><tr><td>2</td><td>16 Pixel, feste breite (7Zeilen)</td></tr><tr><td>3</td><td>19 Pixel (6Anzeigezeilen)</td></tr><tr><td>4</td><td>19 Pixel, feste breite (6Zeilen)</td></tr><tr><td>5</td><td>21 Pixel (5Anzeigezeilen)</td></tr><tr><td>6</td><td>21 Pixel, feste breite (5Zeilen)</td></tr></table> ! **Achtung:** Die in der Tabelle angegebenen Pixelwerte und Zeilen sind nur ungefähre Angaben und können je nach verwendetem Gerät variieren.

Zum Darstellen einer Nachricht auf dem Display des empfangenden Terminals fügen Sie in der HTTP-Response den Parameter

```
df_msg=<Nachricht>,<Dauer>,<Beep>,<Font>
```

ein, also beispielsweise

```
df_msg=Dieses\rist\reine\rNachricht,5,1,0
```

für eine Nachrichten-Ausgabe von „Dies ist eine Nachricht“ (Jedes Wort in einer eigenen Zeile), die für 5 Sekunden dargestellt wird und einen Ok-Signalton auslöst.

! **Achtung:**
Bitte achten Sie beim Text auf die Regeln zur Repräsentation von ‚=‘, ‚?‘, ‚&‘ oder Komma durch die entsprechenden Entitites.

2.2.3.2.5. Icon für die Darstellung einer Nachricht setzen (df_msg_icon)

Parametername	Bedeutung
Nummer der F-Taste	Mit dieser Nachricht können Sie ein Icon für die Darstellung einer Nachricht am Display definieren. Hierbei wird die Nummer der F-Taste übergeben, der das Icon im Design zugeordnet ist.

Beispiel:

```
df_msg_icon=2
```

Stellt für die Ausgabe von Nachrichten das Icon der Kette F2 (typischer Weise „Gehen“) ein.

```
df_msg=Nachricht\rmit%20Icon,5,1,0&df_msg_icon=4
```

Setzt für diese (und eventuell folgende) Nachrichten das Icon auf das der Taste F4.

2.2.3.2.6. Backlight setzen (df_backlight)

Sie können gezielt als Reaktion auf einen Datensatz die Backlights des Geräts einstellten. Dazu nutzen Sie die Anweisung `df_backlight` im HTTP-Response.

Parametername	Bedeutung
Backlight-ID	0: Transponder 1: Logo 2: Tastatur (EVO 4.3) 3: Benutzerspezifisch 4: Fingerprint
Delay	Zeitdauer in Sekunden, für die das Backlight geschaltet wird. 0 = dauerhaft
Intensität Rot	Wert zwischen 0 (kein Rotanteil) bis 255 (voller Rotanteil)
Intensität Grün	Wert zwischen 0 (kein Grünanteil) bis 255 (voller Grünanteil)
Intensität Blau	Wert zwischen 0 (kein Blauanteil) bis 255 (voller Blauanteil)
Intensität Weiß	Wert zwischen 0 (kein Weißanteil) bis 255 (voller Weißanteil)

Anmerkung:

- Nicht alle Geräte haben alle Backlights (aktuell hat nur das EVO 4.3 ein Tastatur-Backlight)
- Nicht alle Backlights sind Voll-Farb-Backlights (ebenfalls im EVO 4.3 ist das Tastatur-Backlight nur weiß)

Beispiel:

```
df_backlight=0,5,255,255,0,192
```

stellt das Transponder-Backlight für 5 Sekunden auf ein helles (192) Gelb (255, 255, 0) ein.

2.2.3.2.7. Hintergrund-Nachricht setzen (df_info_msg)

Parametername	Bedeutung
Nachricht	Textnachricht, die am Display angezeigt werden soll. Einen Zeilenumbruch können Sie durch die Angabe von „\r“ erreichen.
Font	Gibt die Zeichensatzgröße und Art an. 0 – Standardschrift 1 – 16 Pixel (7 Anzeigezeilen) 2 – 16 Pixel, feste breite (7 Zeilen) 3 – 19 Pixel (6 Anzeigezeilen) 4 – 19 Pixel, feste breite (6 Zeilen) 5 – 21 Pixel (5 Anzeigezeilen) 6 – 21 Pixel, feste breite (5 Zeilen) ! **Achtung:** Die in der Tabelle angegebenen Pixelwerte und Zeilen sind nur ungefähre Angaben und können je nach verwendetem Gerät variieren.

Zum Darstellen eines Hintergrund-Infotexts auf dem Display des empfangenden Terminals fügen Sie in der http-Response den Parameter

```
df_info_msg=<Nachricht>,<Font>
```

ein, also beispielsweise

```
df_info_msg= Dieses\rist\reine\rNachricht,0
```

! **Achtung:**
Bitte achten Sie beim Text auf die Regeln zur Repräsentation von ‚=', ‚?', ‚&' oder Komma durch die entsprechenden Entities.

2.2.3.2.8. Onlinefunktion der Zutrittskontrolle (df_ac2)

Parametername	Bedeutung
Modul	Der Wert der Zeichenfolge, muss dem Format des Feldes "TM" der "Reader" Liste folgen. Er muss demnach immer 3 Ziffern umfassen.
Master	Id für den RS485 Bus ZK, Beschreibt den ZK-Bus-Strang. RS485 Bus ID 1 RS485 Bus ID 2 usw. Master muss gemeinsam mit Modul eingesetzt werden.
Maske	Wertemaske für die angesprochenen Einheiten. Sie bilden den Wert durch die Summe der Werte. Möchten Sie Beispielsweise die rote LED und das 3 Relais einschalten, dann übergeben Sie als Wert „4 + 32 = **36**". **Wert / Bit** — **Beschreibung** 1 / 1 — Bei gesetztem Bit, wird der Buzzer angesprochen. 2 / 2 — Bei gesetztem Bit, wird die grüne LED angesprochen. 4 / 3 — Bei gesetztem Bit, wird die rote LED angesprochen. 8 / 4 — Bei gesetztem Bit, wird das Relais 1 angesprochen. 16 / 5 — Bei gesetztem Bit, wird das Relais 2 angesprochen. 32 / 6 — Bei gesetztem Bit, wird das Relais 3 angesprochen. 64 / 7 — Bei gesetztem Bit, wird das Relais 4 angesprochen. 128 / 8 — Bei gesetztem Bit, wird die gelbe LED (ab 04.03.16) / das Relais 5 (bis 04.03.15) angesprochen. 256 / 9 — Bei gesetztem Bit, wird das Relais 5 (ab 04.03.16) / 6 (bis 04.03.15) angesprochen. [ab 04.03.16] 512 / 10 — Bei gesetztem Bit, wird das Relais 6 angesprochen. 1024 / 11 — Bei gesetztem Bit, wird das Relais 7 angesprochen. 2048 / 12 — Bei gesetztem Bit, wird das Relais 8 angesprochen. 4096 / 13 — Bei gesetztem Bit, wird das Relais 9 angesprochen. 8192 / 14 — Bei gesetztem Bit, wird das Relais 10 angesprochen. 16384 / 15 — Bei gesetztem Bit, wird das Relais 11 angesprochen. 32768 / 16 — Bei gesetztem Bit, wird das Relais 12 angesprochen.

<table>
<tr><td></td><td>65536 / 17 Bei gesetztem Bit, wird das Relais 13 angesprochen.
131072 / 18 Bei gesetztem Bit, wird das Relais 14 angesprochen.

[ab 04.03.18]

… / 19 Bei gesetztem Bit, wird das Relais 15 angesprochen.
… / 20 Bei gesetztem Bit, wird das Relais 16 angesprochen.
… / 21 Bei gesetztem Bit, wird das Relais 17 angesprochen.
… / 22 Bei gesetztem Bit, wird das Relais 18 angesprochen.
… / 23 Bei gesetztem Bit, wird das Relais 19 angesprochen.
… / 24 Bei gesetztem Bit, wird das Relais 20 angesprochen.
… / 25 Bei gesetztem Bit, wird das Relais 21 angesprochen.
… / 26 Bei gesetztem Bit, wird das Relais 22 angesprochen.
… / 27 Bei gesetztem Bit, wird das Relais 23 angesprochen.
… / 28 Bei gesetztem Bit, wird das Relais 24 angesprochen.
… / 29 Bei gesetztem Bit, wird das Relais 25 angesprochen.
… / 30 Bei gesetztem Bit, wird das Relais 26 angesprochen.
… / 31 Bei gesetztem Bit, wird das Relais 27 angesprochen.
… / 32 Bei gesetztem Bit, wird das Relais 28 angesprochen.</td></tr>
<tr><td>Status / Funktion</td><td>Zustand, der von den angegebenen Einheiten anzunehmen ist.<table><tr><td>0</td><td>Aus</td></tr><tr><td>1</td><td>Ein</td></tr><tr><td>2</td><td>Wechsel (600ms an, 600ms aus)</td></tr><tr><td>3</td><td>3 mal einschalten für 500ms</td></tr></table></td></tr>
<tr><td>Dauer</td><td>Ist eine Zeitdauer und nur bei Status / Funktion = 1 gültig.
Bedeutung:
- 0 = ständig ein
- 1 - 40 = Dauer in Sekunden ein, für die die Maske aktiv ist.</td></tr>
</table>

Zum Schalten der Zutrittskontrolle fügen Sie den Parameter

```
df_ac2=<Modul>,<Master>,<Maske>,<Status>,<Dauer>
```

in die http-Response ein.

2.2.3.2.9. Custom-Nachricht an ZK-Teilnehmer senden (df_custom_msg_ac2) [in Vorbereitung]

Um Daten direkt an einen Teilnehmer im ZK-Bus zu senden, kann eine df_custom_message_ac2 genutzt werden. Diese wird vom ZK-Controller entgegengenommen und direkt an den dort spezifizierten Teilnehmer weitergeleitet.

Um auch Binärdaten an den ZK-Teilnehmer übermitteln zu können, können die Daten während der HTTP-Übertragung hexadezimal kodiert werden. Wird diese Kodierung verwendet, so fasst der ZK-Controller je zwei aufeinander folgende Hex-Zeichen zu einem Byte zusammen – die Folge der daraus resultierenden Bytes wird dann an den ZK-Teilnehmer übermittelt.

Parametername	**Bedeutung**

Modul	Der Wert der Zeichenfolge, muss dem Format des Feldes "TM" der "Reader" Liste folgen. Er muss demnach immer 3 Ziffern umfassen.
Master	Id für den RS485 Bus ZK, Beschreibt den ZK-Bus-Strang. RS485 Bus ID 1 RS485 Bus ID 2 usw. Master muss gemeinsam mit Modul eingesetzt werden.
Funktion	Code für die auszuführende Funktion 1 = Nachricht direkt an Teilnehmer durchreichen
Kodierung	0 = ASCII (URL kodiert) 1 = HEX
Daten	Daten gemäß Kodierung

Beispiel

```
df_custom_msg_ac2=010,1,1,1,48616c6c6f2057656c74
df_custom_msg_ac2=010,1,1,0,Hallo%20Welt
```

sendet den Text „Hallo Welt“ an den ZK-Teilnehmer 010 im Bus 1.

2.2.3.2.10. Online-Zutrittskontrolle mit Vorprüfung (df_ao_ac2)

Im Kontext der Online-ZK mit Vorprüfung („Assisted Online“) berechnet der ZK-Controller die Aktion, die er im Offline-Betrieb ausführen würde. Diese wird an den Server übermittelt und dessen Zustimmung / Ablehnung erwartet.

Der Datensatz der ZK wird analog zu den Datensätzen, die die ZK aktuell als Buchungsdatensatz erzeugt, generiert. Es wird eine Antwort in folgender Form erwartet:

Parametername	**Bedeutung**
Mode	Entscheidung des Servers. • 0 = Aktion ablehnen • 1 = Aktion annehmen
Group	ID der Gruppe. Falls in der Aktion2-Liste eine Gruppen-Id zur Öffnung benötigt wird und die Ausweis-Id sich nicht in der Identification-Liste befindet.

Beispiele:

```
df_ao_ac2=1
df_ao_ac2=0,1234
```

2.2.3.2.11. ZK-Ereignis durch Server auslösen (df_trigger_ac2)

Mit diesem Kommando kann eine ZK2-Aktion ausgelöst werden, als wäre diese durch einen Leser und einen Nutzerausweis erfolgt. Dazu wird die Leser-Kennung sowie der zu interpretierende Ausweis an das Master-Gerät übermittelt, an dem der Leser angeschlossen ist, die Kodierung des Lesers entspricht der Kodierung in der ZK2-Liste Reader.

Parametername	Bedeutung
Zutrittsmaster	Beschreibt den Verwendeten Busstrang, entspricht der ZM-Spalte in der Reader Liste
Türmodul	Adresse des Lesers im Bus. Entspricht der ZM TM Spalte in der ZK
Ausweis	Der vom Leser simuliert gelesene Ausweis
Pin	Pin für die Zutrittsbuchung entsprechen des Pin Feldes in der Liste Identification

Beispiel:

```
df_trigger_ac2=1,011,876543210,0
```

2.2.3.2.12. Key-Value-Paare durch das Gerät übermitteln (df_kvp)

Viele der Daten, die auf den Geräten vorliegen, werden nicht in Datensatz-Form bereitgestellt. Um an diese Informationen zu gelangen, kann der Webserver per „df_kvp“ das Gerät auffordern, ein oder mehrere Key-Value-Paare zu übermitteln. Auf diesem Weg ist der direkte Zugriff auf

- das aktuelle Datum und die aktuelle Uhrzeit [time],
- die Seriennummer [serialnumber],
- die Firmware-Version [firmwareversion],
- Setup-Variablen und globale Variablen [var],
- Zustände von Relais [relais],
- den Zustand des Flashs [flashstate] und
- Konfigurations-Informationen [info]
- Geräte-Hardware-Informationen [hw]

möglich.

Der Aufbau des „df_kvp“-Kommandos erfordert das übermitteln eines Tokens, das die zu übermittelnden Werte beschreibt (in eckigen Klammern oben aufgeführt). Einige der Tokens erfordern weitere Parameter:

Token-Name	Parameter	Resultat (Beispiel)
firmwareversion	---	`kv=firmwareverssion,\` `  04.03.15.05.EVO35`
flashstate	---	
info		Folgende Informationen werden übermittelt: - Seriennummer `kv=serialnumber,1234` - Gerätetyp, `kv=device,11` - Setup, `kv=setup <fn>,<CRC>` - Zertifikaten `kv=cert <fn>,<CRC>` CRC als 32-Bit CRC in hexadezimaler Schreibweise.

		Siehe Anhang C.2 für Details der CRC-Implementierung.
relais	ID des zu übermittelnden Relais, ab 1 gezählt. Falls nicht angegeben, werden die Zustände aller Relais übergeben.	`kv=relais1,open`
serialnumber	---	`kv=serialnumber,1234`
time	---	`kv=time,2020-09-17T07:00:00`
var	Name der Variablen	`kv=var Ausweis,876543210`
hw	---	`kv=board,50006,4.7a&` (MB) `kv=module,29,1.5c,14&` (PoE) `kv=module,37,1.4c,6&` (Ser) `kv=module,102018,1.2b,6.1&` (TP) `kv=module,11,1.5a,8&` (Eth) `kv=module,30,1.0b,11&` (Gfx) `kv=module,110003,1.1c,11.1` (Dis) Art.Nr. Version Platz

extinfo
 	
extinfo,ac, \
<modul>, \
<master>
	---	`kv=serialnumber,8675&kv=device,23&` (info)
		`kv=setup EVO 2.8.aes,0x12345678&`
		`kv=cert test.cert,0x087654321&`
		
		`kv=firmwareversion,04.03.15.05.EVO35&` (fw)
		
		`kv=board,50006,4.7a&` (hw)
		`kv=module,29,1.5c,14&`
		`kv=module,37,1.4c,6&`
		`kv=module,102018,1.2b,6.1&`
		`kv=module,11,1.5a,8&`
		`kv=module,30,1.0b,11&`
		`kv=module,110003,1.1c,11.1`

Beispiel (Anfrage):

```
df_kvp=serialnumber&df_kvp=var,Ausweis&df_kvp=relais1&df_kvp=relais2
```

Die ermittelten Resultate werden als GET-Request an den Server übertragen. Dieser Request spezifiziert keinen df_table sondern setzt den Request-Typ df_type=kvp und übermittelt im Folgenden key- und value-Teile zu den Anfrage-Parametern als komma-getrennte Paare.

Die Antwort zum obigen Beispiel stellt sich somit wie folgt dar.

URL-Teile	Bemerkung
`http://<host>:<port>/<base-url>?`	
`df_api=1&`	
`df_type=kvp&`	Kein Datensatz – Key-Value-Paare!
`kv=serialnumber,2045&`	Seriennummer
`kv=var Ausweis,876543210&`	Globale Variable: „Ausweis“
`kv=relais1,open&`	Relais 1 geöffnet

`kv=relais2,closed`	Relais 2 geschlossen

2.2.3.2.13. Nicht-ZK-Relais schalten (df_set_relay)

ZK-Relais können mittels df_ac2 im Online-Betriebsmodus der ZK explizit geschaltet werden. Für nicht durch die Zutrittskontroller verwaltete Relais steht der Befehl df_set_relay zur Verfügung, mit dem Relais geschaltet werden können.

Parametername	Bedeutung
Relais-ID	ID des zu schaltenden Relais, ab 1 gezählt
Zustand nach Schaltung	Hier werden die Kodierungen „close“ oder „open“ akzeptiert.
Dauer	Zeitdauer in Sekunden (1 bis 60 in Sekunden, 0 für dauerhaft)

Beispiel:

`df_set_relay=2,close,5` (schließt Relais 2 für 5 Sekunden, danach wird es geöffnet)

2.2.3.2.14. Nicht-ZK-Relais umschalten (df_toggle_relay)

Analog zum expliziten Schalten von Relais können diese – optional für eine gewisse Zeit – umgeschaltet werden.

Parametername	Bedeutung
Relais-ID	ID des zu umschaltenden Relais, ab 1 gezählt
Dauer	Zeitdauer in Sekunden (1 bis 60 in Sekunden, 0 für dauerhaft)

Beispiel:

`df_toggle_relay=1,2` (schaltet Relais 1 für eine Dauer von 2 Sekunden um)

2.2.3.2.15. Dateiübertragung von Server zum Gerät (df_load_file)

Mit dieser Funktion wird eine Transfer-Datei gemäß Kodierung in Anhang A durch das Gerät heruntergeladen. Die Datei wird auf das Gerät übertragen und auf Korrektheit und Anwendbarkeit auf dem Gerät geprüft.
Ist diese Prüfung nicht erfolgreich, so werden alle Daten innerhalb der Transferdatei ignoriert und nicht auf das Gerät angewendet. Es werden entsprechende Systemmeldungen erstellt – sofern diese im Gerätesetup aktiviert sind (vgl. Abschnitt 2.3.1)

Beispiel:

```
df_load_file=/path/on/server?with=optional&server=parameters
```

Hinweis:
Es gibt – wie in Anhang A.3.2 beschrieben – verschiedene Typen der IFF-Datei. Diese entscheiden darüber, wie die Datei durch das Gerät angewendet bzw. verarbeitet wird. Sollten Sie eine Datei ins Flash-Filesystem des Geräts überragen, nutzen Sie bitte den Typen 0xDF00.

Hinweis:
Wenn Sie – bezogen auf einen Request – nur eine IFF-Datei übermitteln wollen, so können Sie diese auch direkt anstelle der Antwort senden (vgl. 2.2.1.2).

Achtung:
Mit dieser Funktion kann natürlich auch das Dateisystem des Geräts missbraucht werden. Ist das Dateisystem des Geräts voll, kann dieses Auswirkungen auf die Funktion des Geräts haben und das Gerät unbrauchbar gemacht werden.

2.2.3.2.16. Dateiübertragung vom Gerät zum Server (df_send_file)

Mit dieser Funktion können Sie gezielt Dateien vom Gerät zum Server übertragen lassen.

- Das Ziel der Übertragung ist die Upload-Form, die als erster Parameter von df_send_file übergeben wird.
- Die zu übertragenden Dateninhalte werden dabei über Tokens angefordert, weitere Parameter eventuell als Komma-getrennte Liste angehängt.

Parametername	**Bedeutung**
Ziel-Path	Pfad auf dem Server, an den die Daten übertragen werden sollen
Token für Daten-Inhalt	Siehe folgende Tabelle mit den Tokens
Optionale weitere Parameter	Weitere Parameter, wie in der Token-Tabelle definiert

Token-Name	**Parameter**	**Beschreibung**
finger	<PID>,<FID> oder <PID>,all oder all	Überträgt ein einzelnes Fingertemplate, die Templates einer Person oder alle auf dem Gerät gespeicherten Fingerprint-Templates. **Hinweis:** Diese Funktion ist aktuell nur für den Flächensensor „Saturn 01“ verfügbar.
finger2	<format>,<PID>,<FID> <format>,<PID>,all oder <format>,all	Analog zu „finger“, <format> darf entweder - 1 (0xDF0E) oder - 2 (0xDF18) sein.
flash	<Dateiname>	Liest eine Datei aus dem Dateisystem des Geräts aus und übermittelt diese an den Server

list	setup,<Listenname> ac2,<Listenname>	Übermittelt die Liste <Listenname>, die in Setup oder Zutrittskontrolle genutzt wird, an den Server.
syslog	<restore-flag>	Übermittelt das Systemlog des Geräts an den Server. Ist das <restore-flag> auf 1 gesetzt, so wird das vollständige Log des Geräts gelesen, wird es als 0 übermittelt, so nur die noch nicht quittierten Datensätze des Logs übermittelt. **Hinweis:** Das Systemlog wird auf dem Gerät zunächst erstellt und dann überragen. Sollten Sie das Systemlog länger nicht ausgelesen habe, kann dieser Prozess etwa 2 Minuten in Anspruch nehmen. Eine Syslog-Upload-Datei kann einen Größe von bis zu 512 kB erreichen.
setup	---	Übermittelt das über http zuvor auf das Gerät übertragene Setup an den Server.
structure	record setup ac2	Übermittelt die Datensatz- bzw. Listen-Beschreibung des aktuellen Setups an den Server.
dir	user root *image [noch nicht implementiert]*	Übermittelt eine Datei mit den absoluten Pfaden aller Dateien im Gerätespeicher. Die einzelnen Pfade werden durch \r\n getrennt.
file *image*	*All [noch nicht impl.]* single *single,<Bildname> [noch nicht impl.]*	*Sende Bilder oder großen Barcode Inhalt:* *Alle Dateien werden gesendet [Noch nicht implementiert].* Die nächste Datei (in chronologischer Reihenfolge) wird gesendet. *Die Datei mit dem Namen <Bildname> wird übertragen [Noch nicht implementiert].*
hip	---	*[Noch nicht implementiert]* *Übermittelt die auf dem Gerät gespeicherten HIP Daten verschlüsselt an den Server.*
language		*[Noch nicht implementiert]*
network		*[Noch nicht implementiert]*

Beispiele:

```
df_send_file=/upload-form.html,flash,root:datafox.cert
df_send_file=/setup-storage-form.pl,setup
df_send_file=/fingerprint-backup-form.asp,finger,1980,all
df_send_file=/list-data-form.js,list,ac2,Identification
df_send_file=/device-logs-form.cgi,syslog,0
df_send_file=/list-desc/,structure,record
```

2.2.3.2.17. Datei auf Gerät löschen (df_remove_file)

Mit dieser Funktionen können Dateien auf dem Gerät gelöscht werden. Der Pfad zu der Datei wird als Parameter übergeben.

Parametername	Bedeutung
Path	Gibt den Pfad für eine Datei im Speicher des Gerätes an. Der Pfad muss mit „user:“ oder „root:“ beginnen. Im Anschluss folgt der Dateiname. **Hinweis:** Die Trennung des Dateisystems in user: und root: wurde mit Firmware 04.03.14 aufgehoben. Ab dieser Release ist daher die Angabe von root: und user: optional!

Beispiele:

```
df_remove_file=root:datafox.cert      (Firmware < 04.03.14)
df_remove_file=datafox.cert                    (Firmware >=
04.03.14)
```

2.2.3.2.18. Löschen von Fingertemplates auf dem Gerät (df_remove_finger)

Das Löschen von Fingern im eingebauten Fingerprint-Leser des MasterIV-Geräts können Sie mit der Anweisung df_remove_finger veranlassen. Hier haben Sie die Möglichkeit, alle im Modul gespeicherten Finger, die Finger einer Person oder einen einzelnen Finger löschen zu lassen.

Dazu übermitteln Sie folgende Anweisungen an das Terminal:

Instruktion	Beschreibung
df_remove_finger=<PID>,<FID>	Löschen des Fingers <FID> der Person <PID>
df_remove_finger=<PID>,all	Löschen aller Finger der Person <PID>
df_remove_finger=all	Löschen aller Finger im Modul

Beispiele:

```
df_remove_finger=1980,all
df_remove_finger=1980,6
```
(Finger 6 ist der rechte Daumen)

Hinweis:
Diese Funktion ist aktuell nur für den Flächensensor „Saturn 01“ verfügbar.

2.2.3.2.19. Komplette Listen aktualisieren (df_setup_list bzw. df_ac2_list)

Die Dateien der Listen werden über denselben HOST-Namen abgerufen, an welchen die Anfrage gestellt wurde.

Achtung:
Die bereitgestellten Listendaten **müssen entsprechend sortiert sein**, wenn Sie im Setup mit Sortierspalte angegeben werden.
Eine ggf. unsortierte Liste würde durch den Client übernommen und als unsortierte Liste gehandhabt, was natürlich zu eventuell langen Suchzeiten führen kann.

Parametername	Bedeutung
Name	Name der Listenbeschreibung.

Zugriffspfad	Pfadangabe inkl. Dateiname der zu ladenden Datei.
	
	Beispiel:
	`df_setup_list=Personalstamm,/pfad/zur/liste.txt`
	oder
	`df_ac2_list=Identification,/pfad/zur/liste.txt`

Bitte achten Sie darauf, dass Listen des Setups und Listen der Zutrittskontrolle gleich heißen dürfen und Sie durch Vorgabe des richtigen Parameter-Namens vorgeben, in welchen Bereich die Liste zu laden ist.

2.2.3.2.20. Datensätze in einer Tabelle zählen (df_table_count)

Liefert die Anzahl der Datensätze, die in der Tabelle enthalten sind. Diese werden unterschieden in die

- Gesamtanzahl der Tabellenzeilen, die nicht gelöscht sind,
- die Anzahl der unsortiert angefügten Zeilen und
- die Anzahl der gelöschten Zeilen.

Parametername	Bedeutung
Tabellen-/Listenname	LIST.<name> oder ACCESS.<name>. Listen mit „LIST“-Präfix werden als im Setup definierte Listen betrachtet, solche mit „ACCESS“-Präfix als Listen der Zutrittskontrolle.

Als Resultat wird ein KVP-Datensatz (siehe auch 2.2.3.2.12) mit folgendem Aufbau geliefert:

URL-Teile	Bemerkung
`http://<host>:<port>/<base-url>?`	
`df_api=1&`	
`df_type=kvp&`	Kein Datensatz – Key-Value-Paare!
`kv=table,list.PID&`	Tabelle, deren Statistikdaten geliefert werden
`kv=count,220&`	Die Tabelle enthält 220 ungelöschte Zeilen
`kv=appended,12&`	An die Tabelle wurden 12 Zeilen unsortiert angefügt

`kv=deleted,18`	Innerhalb der Tabelle befinden sich 18 gelöschte Zeilen

Hinweis:
Die Zugriffsperformance auf eine Tabelle wird schlechter, je mehr gelöschte oder unsortiert angefügte Zeilen enthalten sind. Sofern die Zugriffszeiten auf eine Tabelle nicht ausreichen sollten, sollte diese gelesen und zurückgeschrieben werden. Durch diesen Ablauf wird sichergestellt, dass die Tabelle sortiert im Gerät abgelegt wird und damit die Zugriffszeiten über Schlüsselspalten optimiert werden.

Achtung:
Die bereitgestellten Listendaten **müssen entsprechend sortiert sein**, wenn Sie im Setup mit Sortierspalte angegeben werden.
Eine ggf. unsortierte Liste würde durch den Client übernommen und als unsortierte Liste gehandhabt, was natürlich zu eventuell langen Suchzeiten führen kann.

2.2.3.2.21. Auswahl aus einer Tabelle (df_table_select)

Mit diesem Befehl wird eine Liste bzw. ein Teil einer Liste ausgelesen und an den Server gesendet.

Beispiele:

```
df_table_select=list.PID
df_table_select=access.IDENTIFICATION
```

Parametername	Bedeutung
Tabellen-/Listenname	LIST.<name> oder ACCESS.<name>
Ziel-Path	Pfad auf dem Server, an den die Daten übertragen werden sollen
Erstes Filterkriterium (opt.)	<Spaltenname>=<Wert>
Zweites Filterkriterium (opt.)	<Spaltenname>=<Wert>

Beispiele (die Tabelle PID besteht aus den vier Spalten PID, Name, Abteilung, Datum):

```
df_table_select=list.PID,/upload/form
df_table_select=list.PID,/upload/form,Abteilung%3dEntwicklung
df_table_select=list.PID,/upload/form,Abteilung%3dEntwicklung,PID%3d5
```

Die Selektionen liefern

- alle Datensätze, die in der Tabelle PID enthalten sind,
- alle Datensätze, in deren Abteilung-Spalte der Text „Entwicklung“ steht
- alle Datensätze, in deren Abteilung-Spalte der Text „Entwicklung“ steht und deren PID 5 ist

2.2.3.2.22. Anfügen an eine Tabelle (df_table_append)

Anfügen eines Datensatzes an eine Tabelle. Bei sortierten Tabellen wird dieser Datensatz im unsortierten Bereich der Tabelle angefügt. Es ist erforderlich, dass alle Felder der Tabelle als komma-getrennte Liste in der Instruktion übergeben werden.

Parametername	Bedeutung
Tabellen-/Listenname	LIST.<name> oder ACCESS.<name>
Datenfelder	Liste der Felder, die an die Tabelle angefügt werden soll.

Beispiele (die Tabelle PID besteht aus den vier Spalten PID, Name, Abteilung, Datum):

```
df_table_append=list.PID,5,Sven%20Meyer,Entwicklung,
df_table_append=list.PID,9999,,Besucher,
```

Als Resultat des Anhängens wird eine Systemmeldung erzeugt, die über Erfolg oder die Fehlerursache beim Anfügen informiert.

2.2.3.2.23. Ändern der Daten einer Tabelle (df_table_update)

Parametername	Bedeutung	
Tabellen-/Listenname	LIST.<name> oder ACCESS.<name>	
Erstes Filterkriterium	<Spaltenname>=<Wert>	
Zweites Filterkriterium	<Spaltenname>=<Wert>	
Erste, zu ändernde Spalte	<Spaltenname>=<Wert>	Werte aus maximal 4 Spaten können so geändert werden.
Weitere, zu ändernde Spalten (opt.).	<Spaltenname>=<Wert>	

Beispiele (die Tabelle PID besteht aus den vier Spalten PID, Name, Abteilung, Datum):

```
df_table_update=list.PID,,,Abteilung%3d
df_table_update=list.PID,Abteilung%3dEntwicklung,,Abteilung%3dDevelopme
nt
df_table_update=list.PID,Abteilung%3dEntwicklung,PID%3d5,Datum%3d2019-
09-06T06:23:00
```

Die erste Instruktion löscht in allen Einträgen der Personalliste den Wert der Spalte Abteilung.

Die zweite Instruktion ändert alle Auftreten von „Entwicklung“ in der Abteilungsspalte in „Development“ ab.

Die dritte Instruktion ändert den Wert der Datum-Spalte des Mitarbeiters mit PID 5 aus Abteilung Entwicklung auf den 06.09.2019 um 06:23:00 ab. Es können Werte in 4 Spalten durch ein df_table_update Statement geändert werden.

Achtung:
Intern wird das Ändern einer Zeile dadurch realisiert, dass die selektierten Zeilen als gelöscht markiert und neue Zeilen angefügt werden.

2.2.3.2.24. Löschen von Daten aus einer Tabelle (df_table_delete)

Parametername	Bedeutung
Tabellen-/Listenname	LIST.<name> oder ACCESS.<name>
Erstes Filterkriterium (opt.)	<Spaltenname>=<Wert>
Zweites Filterkriterium (opt.)	<Spaltenname>=<Wert>

Beispiele (die Tabelle PID besteht aus den vier Spalten PID, Name, Abteilung, Datum):

```
df_table_delete=list.PID
df_table_delete=list.PID,Abteilung%3dEntwicklung
df_table_delete=list.PID,Abteilung%3dEntwicklung,PID%3d5
```

Analog zum Befehl `df_table_select` löscht diese Instruktion Datensätze, die mit `df_table_select` ausgewählt werden können. Diese sind

- alle Datensätze, die in der Tabelle PID enthalten sind,
- alle Datensätze, in deren Abteilung-Spalte der Text „Entwicklung“ steht
- alle Datensätze, in deren Abteilung-Spalte der Text „Entwicklung“ steht und deren PID 5 ist

2.2.3.2.25. Firmware-Update über HTTP(S) (df_load_firmware)

Das Kommando “df_load_firmware” ist einer von drei Wegen, auf denen ein Firmware-Update vom HTTP-Server zum Gerät übermittelt werden kann (vgl. Anhang E)

Dazu übermittelt der Webserver an das Gerät, aus welchem Firmware-Paket welche Firmware-Datei bezogen werden soll.

Beispiele:

```
df_load_firmware=04.03.19.21.dfz,evo3.5_04.03.19.21.iff
df_load_firmware=04.03.19.21.dfz,evo_intera_II_49004_04.03.19.21.iff
```

Das Gerät generiert daraufhin einen HTTP-Request für den über die Systemvariablen `http.update.host`, `http.update.port` und `http.update.send` festgelegten Firmware-Update-Endpoint. Je nach aktueller Gerätkommunikationseinstellung wird hier entweder HTTP oder HTTPS genutzt.

Beispiele:

GET https://update.host:443/update-server-send-path/04.03.19.21.dfz/evo3.5_04.03.19.21.iff

Hinweis:

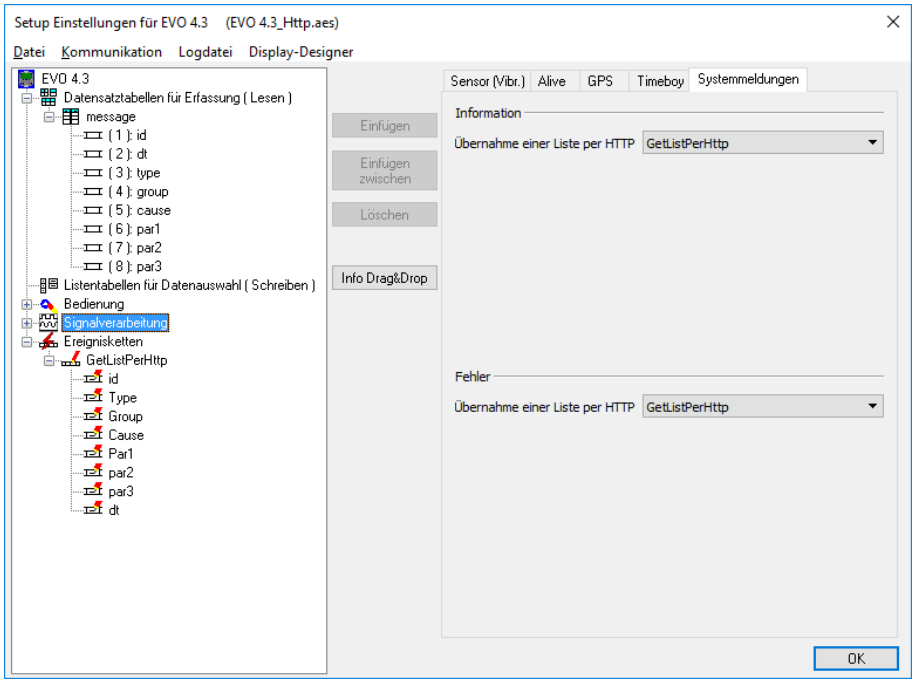
Der angefragte Update-Server ist nun in der Verantwortung, den Inhalt der IFF-Datei aus dem DFZ herauszusuchen und dem Gerät bereitzustellen. Dieses könnte z.B. durch einen „normalen“ Webserver erfolgen, der über die – in dedizierte Ordner – entpackten DFZ-Archive verfügt.

2.3. Quittungen über Systemmeldungen

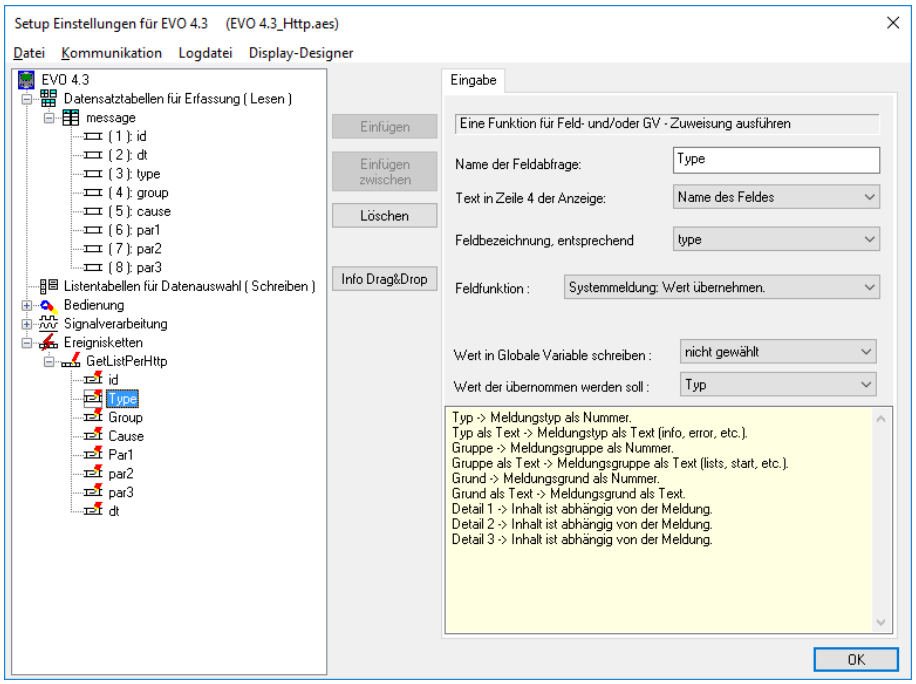
Unabhängig davon, ob sie die DFCom-Schnittstelle oder die HTTP-API einsetzen, können Sie sich vom Gerät die Durchführung von Aktionen quittieren lassen.

Die Quittierung erfolgt durch Systemmeldungen. Dieses Kapitel stellt diese exemplarisch im Kontext der Listendatenüberragung vor.

Systemmeldungen legen sie in dem verwendeten Setup unter Signalverarbeitung an.



Durch die zugewiesene Ereigniskette werden Datensätze erzeugt. Diese stehen Ihrer Anwendung zur Verfügung.



Über die Feldfunktion „Systemmeldung: Wert übernehmen“ können Sie sich die notwendigen Inhalte des Datensatzes zusammenstellen.

Hinweis:
Um zu erkennen, von welchem Gerät die Quittierung kommt, können verschiedene Möglichkeiten verwendet werden.
Hier eine kleine Auswahl:

- Sie können in der Eingabekette den Gerätenamen und die Seriennummer des Gerätes als Id verwenden.
- Sie können über die Response auch eine globale Variable stellen, welche in den erzeugten Systemmeldungen wieder als Feldwert mit gesendet wird. Dieser Weg eine Session-Id zu erstellen, wurde auch in dem obigen Beispiel verwendet.

2.3.1. Systemmeldungen (Feedback – Datensätze)

Zurzeit können folgende Systemmeldungswerte generiert werden:

Grund	Gruppe	Typ	Beschreibung
	Detail 1 / Detail 2 / Detail 3		
Allgemeine Meldungen			
0	100 (http)	1	Operation erfolgreich, z.B. Liste wurde übernommen.
	- / - / -		
1	100	2	Allgemeiner Fehler
	- / - / -		
2	0	1	Letzter Serverbefehl erfolgreich

Grund	Gruppe	Typ	Beschreibung
	Detail 1 / Detail 2 / Detail 3		
	- / - / -		
3	0	2	Letzter Serverbefehl enthielt fehlerhafte Teilebefehle – dazu wurden Einzelmeldungen bereits generiert.
	- / - / -		
Listendatenverarbeitung			
1001	100	2	Ungültiger Parameter
	- / - / -		
1002	100	2	Unbekannte Liste
	Unknown List / <Listenname> / <Listentyp>		
1003	100	2	Parameter fehlt
	- / - / -		
1004	100	2	Fehler in Listenzeile
	<Details zum Fehler> / <Listenname> / -		
1005	100	2	Liste wird ignoriert
	- / - / -		
1006	100	2	Liste doppelt angegeben
	- / - / -		
1007	100	2	Das Listen-Update kann nicht durchgeführt werden, da noch ein aktuelles Update in der Verarbeitung ist. (obsolet, wird nicht mehr verwendet)
	- / - / -		
1008	100	2	Encoding wird nicht unterstützt. In Firmware 04.03.10.xx wird nur „ISO-8859-1“ und binäres Encoding (fehlender Encoding Chunk) unterstützt. (obsolet, wird nicht mehr verwendet)
	- / - / -		
1009	100	2	Fehlerhafte CRC innerhalb der Transfer-Datei (obsolet, wird nicht mehr verwendet)
	- / - / -		
1010	100	2	Der Parameter für den http Upload wird (noch) nicht unterstützt. Der Detail 2 enthält den übergebenen Parameter.
	Command unknown / <Parameter> / -		

Grund	Gruppe	Typ	Beschreibung
	Detail 1 / Detail 2 / Detail 3		
1011	100	1	Die Übermittlung von Zutrittslisten wurde veranlasst, aber die Zutrittskontrolle ist deaktiviert.
	no access control / <Listenname> / <Listentyp>		
1012	100	2	Die IFF-Datei für den Upload konnte nicht erstellt werden. Nähere Infos über Detail 1,2 und 3.
	<Detail1> / <Detail2> / <Detail3>		
1013	100	2	Ein übergebener Parameter ist fehlerhaft. Nähere Informationen sind in Detail 1,2 und 3 zu finden.
	<Detail1> / <Detail2> / <Detail3>		
1014	100	1	Eine Upload Anfrage wurde entgegen genommen und wird jetzt verarbeitet Detail 2 enthält die Anfrage-Details
	Received an upload request / <Dateityp> / -		
1015	100	2	Fehler beim Upload der IFF Datei
	HTTP-Upload failed. Could not connect to server / <Dateiname> / <Pfad>		
1016	100	1	Datei wurde erfolgreich an den Server übertragen. Response OK am Ende der ge-chunk-ten Dateiübertragung.
	HTTP-Upload finished - response ok / <Dateiname> / <Pfad>		
1017	100	2	Datei wurde an den Server übertragen. Das Ok des Servers nach Übermittlung des letzten Chunks fehlt.
	HTTP-Upload finished - response error / <Dateiname> / <Pfad>		
1018	100	2	Während eines Updates darf kein weiteres durchgeführt werden.
	Update already in progress / - / -		
1019	100	2	Listentyp nicht bekannt.
	There are no lists with this type / <Listenname> / <Typ>		
1020	100	2	Liste nicht bekannt.
	Unkown list / <Listenname> / <Typ>		
1021	100	2	Fingerprint, PID ungültig.
	Error reading PID / <PID> / -		
1022	100	2	Fingerprint, FID ungültig.
	Error reading FID / <FID> / -		
1023	100	1	Download der Datei war erfolgreich.
	Download ok / <Dateiname> / <Pfad>		

Grund	Gruppe	Typ	Beschreibung
	Detail 1 / Detail 2 / Detail 3		
1024	100	2	Fehler beim Download der Datei.
	Download error / <Dateiname> / <Pfad>		
1025	100	1	Eine IFF-Datei wurde zur Übermittlung eingeplant
	IFF-file added to upload / <Dateiname> / <Pfad>		
1026	100	2	Speicherproblem, zur Verarbeitung der Anfrage steht nicht genügend Speicher zur Verfügung.
	- / - / -		
1027	100	2	Im Gerät liegt kein gültiges Setup vor. Anmerkung: Das Gerät wurde per `df_send_file` zur Übermittlung des Setups aufgefordert, kann das aber nicht tun, da das Setup nicht per http auf das Gerät übermittelt wurde.
	No valid setup / - / -		
1030	100	1	Download der Datei war erfolgreich (obsolet, wird nicht mehr verwendet)
	HTTP-Download OK / <Name> / <Pfad>		
1031	100	2	Download der Datei nicht erfolgreich (obsolet, wird nicht mehr verwendet)
	HTTP-Download Error / <Name> / <Pfad>		
1032	100	2	Listenübertragung: Zu viele Spalten in einer Zeile.
	Line <LineNo> / <Daten> / <Fehlercode>		
1033	100	2	Listenübertragung: Zu wenige Spalten in einer Zeile.
	Line <LineNo> / <Daten> / <Fehlercode>		
1034	100	2	Listenübertragung: Eine Spalte ist zu breit.
	Line <LineNo> / <Daten> / <Fehlercode>		
1035	100	2	Listenübertragung: Bei Übertragung von HEX-ASCII fehlt ein Zeichen.
	Line <LineNo> / <Daten> / <Fehlercode>		
1036	100	2	Listenübertragung: Bei Übertragung von HEX-ASCII ist ein Nicht-Hex-Zeichen dabei.
	Line <LineNo> / <Daten> / <Fehlercode>		
1037	100	2	Listenübertragung: ZK-Liste wird bei deaktivierter Zutrittskontrolle ignoriert.
	Ignore access list / <Name> / -		

Grund	Gruppe	Typ	Beschreibung
	Detail 1 / Detail 2 / Detail 3		
1038	100	2	Es ist keine Setup Datei auf dem Gerät vorhanden
	- / - / -		
1039	100	2	http Header zu groß
	http Header too large / <headerSize> / <maxSize>		
Verarbeitung von IFF Dateien			
1501	150	2	Fehler beim Lesen von Daten. Genauere Informationen sind den Details 1 und 2 zu entnehmen.
	<Fehlerbeschreibung> / <Details zum Fehler> / -		
1502	150	2	Fehler beim Schreiben der Daten. Genauere Informationen sind den Details 1 und 2 zu entnehmen.
	<Fehlerbeschreibung> / <Details zum Fehler> / -		
1503	150	2	Das Setup passt nicht zum Gerätetypen und wird daher vom Gerät nicht übernommen.
	AES file not valid on this device / <Name des Gerätetyps> / -		
1504	150	2	Fehler bei der CRC Prüfung
	<Typ der IFF Datei> / - / -		
1505	150	2	Versionierung im IFF Chunk wird nicht unterstützt.
	<Typ der IFF Datei> / - / -		
1506	150	2	Das gewählte Encoding wird von der Firmware nicht unterstützt.
	<Typ der IFF Datei> / - / -		
1507	150	2	Fehlerhaftes Encoding der IFF Datei.
	<Typ der IFF Datei> / - / -		
1508	150	2	Speicherfehler beim Erstellen einer IFF Datei
	<Typ der IFF Datei> / - / -		
Routing			
1800	180	1	Eintrag zum Anwenden gefunden
	Routing destination found / <Name der Routing Datei> / <Adresszusatz>		
1801	180	1	Eintrag zum Weiterleiten gefunden
	Routing forward file / <Name der Routing Datei> / <Empfänger Adresse>		
1802	180	1	Routing Datei empfangen
	Routing info received / <Name der Routing Datei> / -		

Grund	Gruppe	Typ	Beschreibung
	Detail 1 / Detail 2 / Detail 3		
1803	180	1	Es gibt noch Einträge, die bei der letzten Verarbeitung nicht abgearbeitet wurden.
	Routing entries left / <Name der Routing Datei> / -		
1804	180	1	Routing Datei fertig abgearbeitet. Datei wird gelöscht
	Routing finished/ <Name der Routing Datei> / -		
1805	180	2	Kein gültiger Eintrag gefunden. Datei wird gelöscht
	No valid entry / <Name der Routing Datei> / -		
1806	180	2	Routing Datei konnte nicht angewandt werden
	Update failed / <Name der Routing Datei> / <Adresszusatz>		
1807	180	2	Fehler beim Weiterleiten der Routingdatei
	Forwarding failed / <Name der Routing Datei> / <Empfänger Adresse>		
Fingerprint			
2001	200	2	Fehler beim Lesen des Fingerprint-Templates
	- / - / -		
2002	200	2	Fehler beim Schreiben des Fingerprint-Templates
	- / - / -		
2003	200	2	Fehler beim Löschen eines Fingerprint Templates.
	Error deleting template / - / -		
2004	200	1	Finger wurde am Sensor eingelernt.
	<PID> / <FID> / <Qualität>		
2005	200	1	Fingertemplate wurde optimiert.
	<PID> / <FID> / -		
2006	200	1	Fingertemplate wurde hinzugefügt.
	<PID> / <FID> / -		
2007	200	1	Fingertemplate wurde gelöscht.
	<PID> / <FID> / -		
2008	200	1	Alle Fingertemplates einer PID wurden gelöscht.
	<PID> / - / -		
2009	200	1	Alle vorhandenen Templates wurden gelöscht.
	- / - / -		
2010	200	2	Fingerprint ist nicht verfügbar.

Grund	Gruppe	Typ	Beschreibung
	Detail 1 / Detail 2 / Detail 3		
	Fingerprint is not available / - / -		
2011	200	2	Nicht kompatible Template Typen in Gerät und IFF-Datei.
	Bad Template Type / < Template Type Device> / <Template Type IFF>		
Filesystem			
2500	250	1	Datei wurde erfolgreich gelöscht.
	File successfully deleted / <Dateiname> / -		
2501	250	2	Datei konnte nicht gefunden werden.
	File not found / <Dateiname> / -		
2502	250	2	Unbekannte Partition (obsolet, diese Meldung wurde nur bis Firmware 04.03.13 generiert)
	- / - / -		
2503	250	2	Fehler beim Löschen der Datei.
	Error while deleting the file / <Dateiname> / -		
Tabellenbefehle			
3000	300	2	Öffnen der Liste beim Append-, Update- oder Delete-Befehl fehlgeschlagen.
	Error opening the list / <Listenname> / -		
3001	300	2	Fehler beim Hinzufügen eines Datensatzes an eine bestehende Liste.
	Error adding list data / <Fehlerhafte Daten> / -		
3002	300	2	Beim Update- oder Delete-Befehl kann der Filter nicht gesetzt werden.
	List selection not possible / <Listenname> / -		
3003	300	2	Fehler in der Parameterliste beim Select- oder Update-Befehl.
	Parameter missing / <Listenname> / -		
3004	300	2	Fehler beim Löschen einer kompletten Liste.
	Error deleting the entire list / <Listenname> / -		
3005	300	2	Interner Speicherfehler
	Memory error / - / -		
Firmware-Update über HTTP(S)			
3500	350	1	Update erfolgreich

Grund	Gruppe	Typ	Beschreibung
	Detail 1 / Detail 2 / Detail 3		
	FW update success / <Neue Firmware Version> / <evtl. Branch-Name>		
3501	350	2	Download der Update-Datei fehlgeschlagen
	Server request failed / <Parameter von `df_update_firmware`> / <http.update.send>		
3502	350	2	Update-Server meldet Fehler bei Verarbeitung der Update-Anfrage.
	<Parameter von `df_update_firmware`> / <http.update.send> / <Fehlercode des Update-Servers>		
3503	350	2	Die Firmware unterstützt ein Modul des Geräts nicht.
	Module not supported / <Modul Id> <Index> / <Name des Moduls>		
3504	350	2	Firmware für anderes Gerät übertragen
	Invalid device type / <Geräte-Typ ID> / <Geräte-Typ ID aus IFF Datei>		
Firmware-Update über HTTP(S) / Verfügbarkeits-µS			
3800	380	1	Firmware-Version ist verfügbar
	<Name DFZ-Datei> / - / -		
3801	380	2	Nicht unterstützter Anfrage-Modus
	unsupported query mode / - / -		
3802	380	2	Keine Firmware gefunden
	no match / - / -		
3803	380	2	Interner Verarbeitungsfehler
	internal error / - / -		
Firmware-Update über HTTP(S) / Kompatibilitäts-µS			
3900	390	1	Gerät ist kompatibel zur Firmware
	<Name DFZ-Datei> / <Name IFF Datei> / -		
3901	390	2	Unbekannter Parameter bei Nutzung des Service
	unhandled parameter / <Parameter> / -		
3902	390	2	Die Anfrage enthielt keinen Gerätetyp
	device type missing / - / -		
3903	390	2	Die Anfrage enthielt keine Seriennummer
	serial number missing / - / -		
3904	390	2	Die Anfrage enthielt keinen Daten zur Hauptplatine

Grund	Gruppe	Typ	Beschreibung
	Detail 1 / Detail 2 / Detail 3		
	board missing / - / -		
3905	390	2	Die Anfrage enthielt keine zu installierende Firmware-Version
	no firmware version specified / - / -		
3906	390	2	Die Anfrage enthielt keine Hardware-Module
	no hardware modules specified / - / -		
3907	390	2	Auf dem Server ist der Firmware kein MD5 Fingerprint zugeordnet
	md5 fingerprint missing / - / -		
3908	390	2	Der MD5-Fingerprint der Anfrage ist falsch
	md5 fingerprint wrong / - / -		
3909	390	2	Verzeichnis mit entpackten Firmware-Dateien kann nicht geöffnet werden.
	directory open error / - / -		
3910	390	2	Keine zum Gerätetyp passende Firmware-IFF-Datei gefunden
	no iff file found / - / -		
3911	390	2	Die Version der Kompatibilitätsinformationen (COMP) stimmt nicht (ist nicht 2).
	no acceptable compatibility info / - / -		
3912	390	2	Die Version des Hilfs-Chunks (FAUX) stimmt nicht (ist nicht 1).
	no acceptable aux info / - / -		
3913	390	2	Der Gerätetyp der Kompatibilitäts-Anfrage stimmt mit keinem Gerätetypen, der in der IFF-Datei enthalten ist, überein.
	device type mismatch / - / -		
3914	390	2	Eine Hardware-Komponenten des Geräts wird in der dort eingesetzten Version nicht durch die Firmware unterstützt.
	unsupported hardware / <fw-idx>,<version>[,<place>] / -		
3915	390	2	Das Verzeichnis mit Firmware-Daten existiert nicht
	firmware directory not existing / <Directory> / -		
3916	390	2	MPU nicht bekannt

Grund	Gruppe	Typ	Beschreibung
	Detail 1 / Detail 2 / Detail 3		
	failed to derive MPU / <Geräte-Typ-Kennung> / -		
Datum / Uhrzeit ändern			
4000	400	1	Datum / Uhrzeit wurde erfolgreich geändert
	Time changed / <Datum/Uhrzeit> / -		
4001	400	2	Fehler im Format des Zeitstrings
	Format error / <Datum/Uhrzeit> / -		
4002	400	2	Interner Verarbeitungsfehler beim Stellen der Uhrzeit.
	Internal error / - / -		
Systemvariablen setzen			
4500	450	1	Die Variable wurde erfolgreich geändert.
	Variable changed / <Name>=<Wert> / -		
4501	450	2	Fehler beim Setzen der Variablen.
	Error when setting the variable / <Name>=<Wert> / <Fehlercode>		
4502	450	2	Fehler beim Lesen der Variablen.
	read error / <Request> / -		
4503	450	2	Fehler beim Lesen der Variablen.
	missing parameter / <Request> / -		
Relais schalten			
5000	500	1	Das Relais wurde erfolgreich geschaltet.
	Relais switched / <Parameter> / -		
5001	500	2	Mind. ein Parameter ist außerhalb seines Gültigkeitsbereiches.
	Parameter error / <Parameter> / -		
5002	500	2	Es wurden zu wenige Parameter übergeben.
	Parameter missing / - / -		
5003	500	1	Das Relais wurde erfolgreich umgeschaltet.
	Relais toggled / <Parameter> / -		
Display-Nachrichten empfangen			
5500	500	1	Eine Nachricht wurde empfangen.
	Message received / <Nachrichtentext> / -		

Grund	Gruppe	Typ	Beschreibung
	Detail 1 / Detail 2 / Detail 3		
5501	500	2	Auf den gesendeten Online-Datensatz wird eine Nachricht erwartet, die nicht vom Server gesendet wurde.
	Message missing / - / -		
5502	500	2	Die neue Nachricht wird nicht angezeigt, weil aktuell noch eine andere am Display dargestellt wird.
	Message ignored / <Parameter> / -		
5503	500	1	Eine Infonachricht wurde empfangen.
	Info message received / - / -		
5504	500	2	Im Setup ist die Funktion „Server Online“ nicht aktiviert. Daher dürfen keine Nachrichten dargestellt werden.
	Online messages disabled / - / -		
5505	500	2	Das df_msg-Kommando wurde als Antwort eines Offline-Datensatzes gesendet.
	Non online message/ - / -		
5510	500	1	df_msg_icon wurde übernommen
	Message icon set / <Dateiname des Icons> / -		
5511	500	2	Die Datei für das Icon wurde nicht gefunden
	Image not found / <filename> / -		
5512	500	1	Ungültige Buzzer-Angabe. Die Nachricht wird ohne Buzzerton angezeigt
	Buzzer invalid / <buzzer> / -		
5513	500	2	Keine gültige Schriftgröße. Die Nachricht wird in der Standard-Größe angezeigt.
	Font invalid / <font> / -		
Akustisches Signal ausgeben			
6000	600	1	Tonfolge bekannt, Signal wird ausgegeben.
	Play Sound Sequence / <Sequenznummer> / -		
6001	600	2	Unbekannte Tonfolge.
	Unknown Sound Sequence / <Sequenznummer> / -		
Eingabekette starten			
6500	650	1	Ereigniskette wird ausgeführt.
	Execute event chain / <EK-Bezeichner> / -		

Grund	Gruppe	Typ	Beschreibung
	Detail 1 / Detail 2 / Detail 3		
6501	650	2	Unbekannte Ereigniskette. Diese kann nicht ausgeführt werden.
	Unknown event chain / <EK-Bezeichner> / -		
Serviceverbindung aufbauen			
7000	700	1	Serviceverbindung wurde aufgebaut.
	Service connection established / <Host:Port> / 0		
7001	700	2	Die Serviceverbindung konnte nicht aufgebaut werden.
	Service connection could not be established / <Host:Port> / <Err-Code>		
Backlights steuern			
7500	750	1	Backlight gestellt.
	Backlight set / <Backlight-Id> / -		
7501	750	2	ID des Backlights ist nicht bekannt
	Unknown backlight / <Backlight-Id> / -		

Verwendete Kodierungen:

Typ:
1: Info – Befehl erfolgreich verarbeitet
2: Fehler

Grund:
Beschreibung der Operation

Gruppe:
100 http-Modul
150 IFF-Verarbeitung
180 Routing
200 Fingerprint-System
250 File-System
300 Listen-/Tabellen-Verarbeitung
350 Firmware-Update über http
380 µService Firmware Verfügbarkeit
390 µService Firmware Kompatibilität
400 Datum / Uhrzeit ändern
450 Systemvariablen ändern
500 Relais schalten
550 Nachrichten anzeigen
600 Audio-Ausgabe
650 Behandlung von Eingabe-Ketten
700 Service Mode
750 Steuerung von Backlights

Eine Quittung für eine erfolgreiche Datenübernahme der ZK-Liste Action erzeugt auf dem Terminal einen Datensatz, der wie folgt übermittelt wird:

URL-Teile	Bemerkung
`http://<host>:<port>/<base-url>?`	
`df_api=1&`	
`df_table=Feedback&`	
`df_col_dt=2017-03-30T13:11:06&`	
`df_col_type=1&`	Typ Info
`df_col_group=100&`	Modul / Gruppe http
`df_col_cause=0&`	kein Fehler
`df_col_par1=action, 4 lines&`	Status-Information
`df_col_par2=Action&`	Name der übernommenen Liste
`df_col_par3=`	

Das Feld `df_col_par3` ist für zukünftige Nutzung reserviert.

2.3.2. Zuordnung von Befehlen und Systemmeldungen

Da die HTTP-Schnittstelle eine asynchrone Datenverarbeitung durchführt, kann der Server in seiner HTTP-Response der dort übergebenen Instruktionen ein „df-action-id“ zuordnen. Diese Action-ID in die der Aktion zugeordneten Systemmeldungen übernommen, so dass die Systemmeldung dem Auslöser zugeordnet werden kann.

Um die Action-ID für einen Befehl zu aktivieren, sendet der Server im HTTP-Response das Header-Feld „df-action-id“ mit. Der zugeordnete Wert darf bis zu 16 Zeichen lang sein (etwa „A39ID“)

Das Gerät greift diesen in der Antwort auf und sendet in der Systemmeldung – ebenfalls im Header – die Action ID sowie eine laufende Nummer (den „df-command-index“), die zum Befehl gehört, mit. Der Command-Index startet bei 0 für das „df_api=1“-Token, der erste Befehl erhält entsprechend den Index 1.

accept	*/*
accept-charset	ISO 8859-1
content-length	664
content-type	application/x-www-form-urlencoded
df-action-id	A39ID
df-command-index	1
host	192.168.123.110:10110
user-agent	Datafox/04.03.19.21.http.16 11.1234

Header einer Systemmeldung

Responses ohne Action ID / Command Index gibt bei den folgenden Systemmeldungen:

- 20xx: Fingerprint-Aktionen in Folge von HTTP-Aktionen
- 3500: Firmware-Update erfolgreich

- Meldungen aus der Zutrittskontrolle

2.4. Verschlüsselung

Achtung:
Im Kontext von API Level 1 auf Hardware 4-Geräten sollten Sie diese Verschlüsselung nicht einsetzen – mit HTTPS ist hier eine deutlich stärkere und standardisierte Technik verfügbar!

Die Datenfelder des Datensatzes können mittels eines Streamchiffre RC4 verschlüsselt werden. Dabei werden die Feldinhalte dann in ihrer Hexadezimaldarstellung übertragen.

Parametername	Bedeutung
df_cb	Der Parameter gibt an, dass alle folgenden Felder bis einschließlich *df_ce* verschlüsselte Feldinhalte haben. Der Wert von *df_cb* enthält den vierstelligen (1000-9999) Public-Key des anzuwendenden Passwortes für den Streamchiffre.
df_ce	Der Parameter gibt an, dass alle folgenden Felder keine verschlüsselten Feldinhalte mehr haben. Wird der Wert korrekt entschlüsselt muss er mit dem Wert von *df_cb* übereinstimmen.

2.4.1. Veranschaulichung der GET-Anfrage

Im Klartext (unverschlüsselt) und verschlüsselt:

Klartext Anfrage
df_api=1&df_record_state=1&df_table=Booking&df_col_sn=2042&df_col_recordtype=1&df_col_badge=3974679390&df_col_timestamp=2017-11-22T08:23:39&df_col_status=online
Klartext Antwort
df_api=1&df_time=2017-11-22T08:24:00
Verschlüsselte Anfrage
df_api=1&df_cb=6102&df_record_state=CC&df_table=66E9B37516AA8C&df_col_sn=0BDC8F79&df_col_recordtype=AB&df_col_badge=AF9B3A929994A5BD7D88&df_col_timestamp=B237B8CA4FA80FD563359C3EE70FE7FC99AF60&df_col_status=9BACFC1E5E0B&df_ce=A344D33B
verschlüsselte Antwort
df_api=1&df_cb=6102&df_time=e1ba6575855619c4d634f7865c01c4b2bc2ec138670ac2&df_ce=a414ebd6

2.4.2. Erkennung einer Verschlüsselung

Um zu erkennen, ob die Datenfelder verschlüsselt versendet werden, wird der Anfang der Verschlüsselung mit ‚df_cb' (Datafox Crypt Begin) gekennzeichnet und mit ‚df_ce' (Datafox crypt end) das Ende gekennzeichnet. ‚df_cb' stellt das erste Feld im Request und ‚df_ce' das letzte Feld im Request dar.

Der Wert des Feldes ‚df_cb' selbst wird im Klartext übertragen und ist der ‚public key'. Er ist eine Zufallszahl zwischen 1000 und 9999. Der Wert muss in Verbindung mit dem Benutzerpasswort für die Ver- und Entschlüsselung herangezogen werden.

Die Chiffrierung der Daten erfolgt somit durch „private key + public key" als Passwortschlüssel.

In der Antwort muss das Feld ‚df_cb' 1:1 zurückgesendet werden. Damit wird sichergestellt, dass die Entschlüsselung erfolgreich war und die Antwort auch zur Anfrage passt.

Der Wert des Feldes ‚df_ce' ist derselbe wie ‚df_cb', wird jedoch verschlüsselt übertragen. Beim Entschlüsseln kann somit sichergestellt werden, ob der verwendete Schlüssel korrekt ist. Der Wert von ‚df_ce' muss daher nach dem Entschlüsseln gleich ‚df_cb' sein.

Gibt es Probleme bei der Entschlüsselung, muss als Antwort ‚df_c=error' gesendet werden. Zusätzlich sind die Felder ‚df_cb' und ‚df_ce' mit Informationen zu bestücken.

Folgende Fehlerfälle sind durch das auswertende Script zu beachten:

‚df_cb' ist keine Zahl oder liegt außerhalb seiner Wertgrenze von 1000 – 9999
- Antwort: df_c=error&df_cb=range&df_ce=unknown/missing
  - Range bedeutet Bereichsfehler, weil der Wert außerhalb seiner Wertgrenzen liegt.
  - Unknown bedeutet unbekannt, weil nicht ermittelt aber verfügbar
  - Missing bedeutet fehlt, es ist keine Angabe in der Anfrage vorhanden.

‚df_cb' ohne abschließendes ‚df_ce'
- Antwort: df_c=error&df_cb=1000&df_ce=missing

‚df_ce' ist keine Zahl oder liegt außerhalb seiner Wertgrenze von 1000 – 9999
- Antwort: df_c=error&df_cb=1000&df_ce=range

‚df_ce' ohne beginnendes ‚df_cb'
- Antwort: df_c=error&df_cb=missing&df_ce=unknown

‚df_ce' ist ungleich ‚df_cb'
- Antwort: df_c=error&df_cb=1000&df_ce=different
  - Different bedeutet ungleich, weil ‚df_ce' nach Entschlüsselung ungleich ‚df_cb' ist.

2.4.3. Rückantwort des WEB-Servers

Die Feldinhalte der Anfrage werden nacheinander mit der RC4 Stromchiffre entschlüsselt. Die Feldinhalte der Rückantwort werden als Teil des Gesamtdatenstroms gesehen und werden im Anschluss an die Entschlüsselung mit der aktuellen Stellung der Stromchiffre wieder verschlüsselt. Einzige Ausnahme ist der erste Feldwert von ‚df_cb'. Dieser wird wie in der Anfrage 1:1, zurückgesendet.
Der Rückantwort muss als letztes verschlüsseltes Feld ‚df_ce' angefügt sein. Der Wert von ‚df_ce' muss gleich dem Wert von ‚df_cb' sein.

2.4.4. Puffergrößen

Die Terminals stellen Puffer in folgender Größe zum Auswerten von http-Antworten des Servers zur Verfügung (Stand 04.03.10.05):

Feld	Puffergröße

http Header	1500 Bytes
http Body	2000 Bytes

Bitte beachten Sie, dass bei einem Überlauf eines der Puffer die Daten abgeschnitten werden und damit die vorgesehene Funktion nicht ausgeführt wird. Insbesondere bei Verwendung von Cookies, die das Gerät nicht auswertet und entsprechend auch nicht zurücksendet, kann der Header-Puffer schnell an seine Grenzen stoßen.

Anhang A: Funktionen von Kommunikations-Bibliothek und Datafox Studio in HTTP Level 1

Dieser Anhang bietet einen Vergleich der Funktionen der Kommunikations-Bibliothek mit der durch die HTTP Schnittstelle bereitgestellten Lösung. Aufgrund der mit dem Active-Mode vergleichbaren Funktionalität der HTTP Schnittstelle werden Funktionen, die sich mit der Passive-Mode Kommunikation befassen, nicht abgebildet.

Legende
Realisiert
Geplant für Release
Geplant für zukünftige Release – noch nicht terminiert.
Nicht geplant zu realisieren – vgl. Kommentar.

Achtung:

Setup-Dateien, die zum Gerät übermittelt wurden, können nur auf demselben Weg zurückgelesen werden. Es ist nicht möglich, Setup-Dateien, die per Kommunikations-Bibliothek-Schnittstelle übermittelt wurden, über HTTP zurückzulesen – und umgekehrt.

A.1: Vergleich von Kommunikations-Bibliothek und HTTP Level 1

Funktionsbeschreibung (aus Kommunikations-Bibliothek-Dokumentation)	Kommunikations-Bibliothek-Funktionsname	HTTP	Status / Plan
Serielle oder TCP/IP für MasterIV initialisieren.	DFCComOpenIV	Nicht erforderlich – Gerät baut Verbindung auf	---
Zuvor geöffnete Schnittstelle schließen.	DFCComClose	Nicht erforderlich – Gerät baut Verbindung auf	---
Erreichbarkeit prüfen.	DFCCheckAE	Nicht erforderlich – Gerät baut Verbindung auf	---
Erreichbarkeit prüfen.	DFCCheckDevice	Nicht erforderlich – Gerät baut Verbindung auf	---
Datum und Uhrzeit schreiben.	DFCComSetTime	df_time (2.2.3.2)	04.03.10.00
Datum und Uhrzeit lesen.	DFCComGetTime	df_kvp=time	04.03.12.01
Anzeigemeldung senden.	DFCComSendMessage	df_msg (2.2.3.2.4)	04.03.10.01
Hintergrundmeldung senden.	DFCComSendInfotext	df_info_msg (analog df_msg, siehe 2.2.3.2.7)	04.03.12.01
Seriennummer lesen.	DFCGetSeriennummer	df_kvp=serialnumber	04.03.12.01
DFCLogOn: Nicht weiter verwenden.	DFCLogOn (veraltet)	Wird nicht umgesetzt.	---
DFCSetLogOn: Protokollierung einschalten.	DFCSetLogOn (veraltet)	Wird nicht umgesetzt.	---

DFCLogOff: Nicht weiter verwenden.	DFCLogOff (veraltet)	Wird nicht umgesetzt.	---
DFCSetLogOff: Protokollierung ausschalten.	DFCSetLogOff (veraltet)	Wird nicht umgesetzt.	---
Rückruffunktion bekannt geben.	DFCSetCallBack	Durch http-Server implementiert	---
DFCSetLogFileName: Name der Protokollierungsdatei setzen.	DFCSetLogFileName (veraltet)	Wird nicht umgesetzt.	---
Fehlernummer zu Fehlertext umsetzen.	DFCGetErrorText	Siehe Systemmeldungen (2.3.1)	04.03.10.03
Wert einer Variablen ändern.	DFCSetGlobVar	df_var (2.2.3.2.2 für globale Variablen, Setup-Variablen und Systemvariablen)	04.03.10.01
Wert einer Variablen ermitteln.	DFCGetGlobVar	df_kvp=var,<VAR_NAME>	04.03.12.01
Relais für bestimmbare Zeit schließen.	DFCCloseRelay	df_set_relay=<RELAIS_ID>,close,duration	04.03.12.01
Relaiszustand abfragen.	DFCGetRelayState	df_kvp=relais,<RELAIS_ID>	04.03.12.01
Relais öffnen.	DFCOpenRelay	df_set_relay=<RELAIS_ID>,open,duration	04.03.12.01
Relais umschalten	---	df_toggle_relay=<RELAIS_ID>,duration	04.03.12.01
Anzahl Ermittlungsversuche ermitteln.	DFCGetDevicePollRetry	Wird nicht umgesetzt.	---
Rückgabe des Schnittstellenhandle.	DFCGetComPort	Wird nicht umgesetzt.	---
Setzen des Schnittstellenhandle.	DFCSetComPort	Wird nicht umgesetzt.	---
In den Kanal schreiben.	DFCWrite	Wird nicht umgesetzt.	---
Aus dem Kanal lesen.	DFCRead	Wird nicht umgesetzt.	---
Gerätedaten (Firmware) laden.	DFCUpload	Transfer von Datei Typ 0xDF01 (Firmware) df_load_file=<PATH ON SRV> Zunächst nicht geplant.	04.03.xx.xx
Version der Gerätesoftware lesen.	DFCGetVersionFirmware	df_kvp=firmwareversion	04.03.12.01
Version der Gerätesoftware lesen.	DFCGetVersionFirmwareFromFile	Nicht möglich über http.	---
Informationen zu einem Modul abfragen.	DFCGetInfo	Zunächst nicht geplant.	---
Transparentmodus einschalten.	DFCOpenComServerMode	Nicht möglich über http.	---
Transparentmodus ausschalten.	DFCCloseComServerMode	Nicht möglich über http.	---
Besteht eine Kommunikationsverbindung.	DFCIsChannelOpen	Nicht möglich über http.	---

Update eines eingebauten Moduls durchführen.	DFCUploadModule	Transfer von Datei Typ 0xDF011 - 0xDF16 (RS9100, Saturn 01, Näherungssensor) Aktuell nicht unterstützt: Biokey 3000/4000/4020, U&Z Funk-Basisstation. df_load_file=<PATH ON SRV>	04.03.12.03
Zusatzoptionen der Firmware lesen.	DFCGetOptionFirmware	Zunächst nicht geplant.	---
Zusatzoptionen der Firmware schreiben.	DFCSetOptionFirmware	Zunächst nicht geplant.	---
Rücksetzfunktion je nach Aufrufmodus.	DFCReset	Kann durch das Setzen der (temporären) Systemvariablen df_var=system.reset,1 ausgelöst werden.	04.03.20.01
Zeichensatztyp für SendMessage und SendInfotext.	DFCSetFontType	Direkt in df_msg (2.2.3.2.4) und df_info_msg (2.2.3.2.7) umgesetzt.	04.03.10.01
Setzen des Passworts für den Zugriffsschutz.	DFCSetPassword	Sicherheit ist durch https implizit gegeben, die Funktion wird daher nicht umgesetzt.	04.03.11.00
Abfragen eines Ersatzschlüssels für das Passwort des Zugriffsschutzes.	DFCGetPasswordKey	Wird nicht umgesetzt.	---
Virtuelles betätigen von Funktionstasten (F1 - F15)	DFCPressVirtualKey	df_ek (siehe 2.2.3.2.3)	04.03.10.00
Aktuellen Zustand des Speicherbausteins abrufen.	DFCGetFlashStatus	df_kvp=flashstate Anmerkung: Hierbei handelt es sich um die Flash-Aufteilung, nicht den Wear-Level-State o.ä. des Bausteins.	04.03.12.01
Setzen des Passworts für die Kommunikationsverschlüsselung.	DFCSetCommunicationPassword	Das Setzen des Passworts für die Kommunikation ist hier nicht möglich – diese Funktion wird durch HTTPS realisiert.	---
Lesen von Informationen zur vorliegenden Gerätehardware und verbaute Optionen.	DFCReadHardwareInfo	Transfer von Datei Typ 0xDF10 df_send_file=<PATH ON SRV>,hip	04.03.xx.xx
Daten einer Datei lesen und auf das Gerät hochladen.	DFCFileUpload	Bitte nutzen Sie spezialisierte Funktionen, wie etwa df_setup_list oder df_ac2_list, falls möglich. Sollten Sie andere Dateien zum Geräte übertragen wollen, die durch diese Funktionen nicht verarbeitet werden, so können diese per df_fs_load übertragen werden: df_fs_load=root/user:<FILENAME ON DEVICE>,<PATH ON SRV>	04.03.12.01
Daten vom Gerät runterladen und in einer Datei speichern.	DFCFileDownload	df_send_file=<PATH ON SRV>,flash,root/user:<File1>…	04.03.12.01
Die zuletzt aufgetretene Fehlernummer ermitteln.	DFCGetLastErrorNumber	Nicht möglich über http.	---
Setupdatei zum Gerät übertragen.	DFCSetupLaden	Transfer von Datei Typ 0xDF02 (Setup Datei) df_load_file=<PATH ON SRV>	04.03.12.01

		Anmerkung: Gerät legt eine Kopie des Original-Setups ab und passt es dann für die Benutzung im Gerät an.	
Setupdaten aus Gerät in Datei schreiben.	DFCDownload	df_send_file=<PATH ON SRV>,setup Anmerkung: Übermittelt Kopie des Setups. Ein per Kommunikations-Bibliothek übertragenes Setup kann nicht per http zurückgelesen werden!	04.03.12.01
Setupdaten in Setupdatei verändern.	DFCModifyStudioFile	Zunächst nicht geplant.	---
Öffnen einer Tabelle.	DFCOpenTable (veraltet, neu DFCTableOpen)	Wird nicht umgesetzt.	---
Schließen einer Tabelle.	DFCCloseTable (veraltet, neu DFCTableClose)	Wird nicht umgesetzt.	---
Filterkriterium setzen.	DFCSetFilter (veraltet, neu DFCTableSetFilter)	Wird nicht umgesetzt.	---
Aktuelles Filterkriterium lesen.	DFCGetFilter (veraltet, neu DFCTableGetFilter)	Wird nicht umgesetzt.	---
Filterkriterium entfernen.	DFCClearFilter (veraltet, neu DFCTableRemoveFilter)	Wird nicht umgesetzt.	---
Datensatzzeiger verschieben.	DFCSkip (veraltet, neu DFCTableSetCurrentRow)	Wird nicht umgesetzt.	---
Wert eines Datensatzfeldes ändern.	DFCSetField (veraltet, neu DFCTableSetCurrentColumnData)	Wird nicht umgesetzt.	---
Wert eines Datensatzfeldes ermitteln.	DFCGetField (veraltet, neu DFCTableGetCurrentColumnData)	Wird nicht umgesetzt.	---
Öffnen einer Listentabelle.	DFCTableOpen	Wird nicht umgesetzt.	---
Schließen einer geöffneten Listentabelle.	DFCTableClose	Wird nicht umgesetzt.	---
Filterkriterium auf eine Spalte setzen.	DFCTableSetFilter	Wird nicht explizit umgesetzt; implizit Bestandteil von df_table_select, df_table_update und df_table_delete	---
Aktuell gesetztes Filterkriterium lesen.	DFCTableGetFilter	Wird nicht umgesetzt, vgl. DFCTableSetFilter	---
Aktuell gesetztes Filterkriterium entfernen.	DFCTableRemoveFilter	Wird nicht umgesetzt, vgl. DFCTableSetFilter	---
Aktuelle Anzahl der Tabellenzeilen lesen.	DFCTableGetRowCount	df_table_count	04.03.15.01
Aktuelle Zeilennummer lesen.	DFCTableGetCurrentRow	Wird nicht umgesetzt – Bitte entsprechendes Filterkriterium und df_table_select nutzen.	---
Aktuelle Zeilennummer verändern.	DFCTableSetCurrentRow	Wird nicht umgesetzt.	---
Aktuelle Datenzeile überschreiben.	DFCTableSetCurrentRowData	Wird nicht umgesetzt – Bitte entsprechendes Filterkriterium und df_table_update nutzen.	---
Spaltenwert der aktuellen Datenzeile überschreiben.	DFCTableSetCurrentColumnData	Wird nicht umgesetzt – Bitte entsprechendes Filterkriterium und df_table_update nutzen.	---

Setzen des Spaltenwertes aller aktuell dem Filterkriterium entsprechenden Zeilen.	DFCTableSetAllRowsToColumnData	df_table_update	04.03.15.01
Aktuelle Datenzeile lesen.	DFCTableGetCurrentRowData	Wird nicht umgesetzt – Bitte entsprechendes Filterkriterium und df_table_select nutzen.	---
Spaltenwert der aktuellen Datenzeile lesen.	DFCTableGetCurrentColumnData	Wird nicht umgesetzt – Bitte entsprechendes Filterkriterium und df_table_select nutzen.	---
Datenzeile an die Tabelle anfügen.	DFCTableAppendRowData	df_table_append	04.03.15.01
Aktuelle Datenzeile löschen.	DFCTableDeleteCurrentRow	Wird nicht umgesetzt – Bitte entsprechendes Filterkriterium und df_table_delete nutzen.	---
Dem Filterkriterium entsprechenden Datenzeilen löschen.	DFCTableDeleteAvailableRows	df_table_delete	04.03.15.01
Datenzeiger im Gerät zurücksetzen.	DFCComClearData	Zunächst nicht geplant.	---
Datenabholvorgang starten.	DFCComCollectData (veraltet, neu DFCReadRecord, DFCQuitRecord)	Das Gerät übermittelt die Daten aktiv.	04.03.12.01
Datensatz ermitteln.	DFCComGetDatensatz (veraltet, neu DFCReadRecord, DFCQuitRecord)	Das Gerät übermittelt die Daten aktiv.	04.03.12.01
Datensatzbeschreibungen aus Gerätesetup ermitteln.	DFCLoadDatensatzbeschreibung	df_send_file=<PATH ON SRV>,structure,datasets	04.03.12.06
Anzahl ermittelter Datensatzbeschreibungen.	DFCDatBCnt	df_send_file=<PATH ON SRV>,structure,datasets	04.03.12.06
Grunddaten einer ermittelten Datensatzbeschreibung erhalten.	DFCDatBDatensatz	df_send_file=<PATH ON SRV>,structure,datasets	04.03.12.06
Grunddaten eines ermittelten Datensatzbeschreibungsfeldes erhalten.	DFCDatBFeld	df_send_file=<PATH ON SRV>,structure,datasets	04.03.12.06
Nächsten anstehenden Datensatz lesen.	DFCReadRecord	Nicht erforderlich, Gerät sendet DS autark	04.03.10.00
Zuvor gelesenen Datensatz quittieren (löschen).	DFCQuitRecord	df_api=1 im Response des Servers	04.03.10.00
Wiederherstellen von Datensätzen.	DFCRestoreRecords	Zunächst nicht geplant.	---
Rohdaten einer Liste importieren.	DFCMakeListe	Nicht erforderlich.	---
Importierte Listendaten in Gerät schreiben.	DFCLoadListen	df_setup_list oder df_ac2_list (2.2.3.2.19)	04.03.10.01
Listenspeicher für Import löschen.	DFCClrListenBuffer	Nicht erforderlich.	---
Listenbeschreibungen aus Gerätesetup ermitteln.	DFCLoadListenbeschreibung	df_send_file=<PATH ON SRV>,structure,lists	04.03.12.06
Anzahl ermittelter Listenbeschreibungen.	DFCListBCnt	df_send_file=<PATH ON SRV>,structure,lists	04.03.12.06
Grunddaten einer ermittelten Listenbeschreibung erhalten.	DFCListBDatensatz	df_send_file=<PATH ON SRV>,structure,lists	04.03.12.06
Grunddaten eines ermittelten Listenbeschreibungsfeldes erhalten.	DFCListBFeld	df_send_file=<PATH ON SRV>,structure,lists	04.03.12.06

Rohdaten einer Liste importieren.	DFCMakeEntranceList	Nicht erforderlich.	---
Importierte Listendaten in Gerät schreiben.	DFCLoadEntranceList	df_setup_list oder df_ac2_list (2.2.3.2.19)	04.03.10.01
Listenspeicher für Import löschen.	DFCClearEntranceListBuffer	Nicht erforderlich.	---
Rohdaten einer Liste importieren.	DFCMakeEntrance2List	Nicht erforderlich.	---
Importierte Listendaten in Gerät schreiben.	DFCLoadEntrance2List	df_setup_list oder df_ac2_list (2.2.3.2.19)	04.03.10.01
Listenspeicher für Import löschen.	DFCClearEntrance2ListBuffer	Nicht erforderlich.	---
Zutrittsprüfung ausführen.	DFCEntrance2Identification (veraltet, neu DFCAccessControlIdentification)	Wird nicht umgesetzt, vgl. df_trigger_ac2 (2.2.3.2.11)	---
Zutrittsmodul ansprechen.	DFCEntrance2OnlineAction (veraltet, neu DFCAccessControlOnlineAction)	Wird nicht umgesetzt.	---
Zutrittsprüfung ausführen.	DFCAccessControlIdentification	df_trigger_ac2=<LESER>,<ID-Code>	04.03.12.06
Zutrittsmodul ansprechen.	DFCAccessControlOnlineAction	df_ac2 (siehe 2.2.3.2.8)	04.03.10.01
Anfügen eines Fingertemplates.	DFCFingerprintAppendRecord / DFCFingerprintRestore	Transfer von Datei Typ 0xDF18 – mit anzuhängenden Templates df_load_file=<PATH ON SRV> Anmerkung: Die Fingertemplates aus der Datei des Servers werden zu denen auf dem Modul gemischt. Auf dem Modul vorhandene Finger mit gleicher PID und FID werden dabei durch den Finger aus der Datei ersetzt.	04.03.12.05 (Flächensensor) 04.03.15.08 (Zeilensensor)
Ermitteln der Daten eines Fingertemplates.	DFCFingerprintGetRecord / DFCFingerprintBackup	df_send_file=<PATH ON SRV>,finger,<PID>,<FID> oder df_send_file=<PATH ON SRV>,finger,<PID>,all oder df_send_file=<PATH ON SRV>,finger,all Erzeugt Transfer-Datei mit Typ 0xDF18	04.03.12.05 (Flächensensor) 04.03.15.08 (Zeilensensor)
Löschen von Fingertemplates.	DFCFingerprintDeleteRecord	df_remove_finger=<PID>,<FID> oder df_remove_finger=<PID>,all oder df_remove_finger=all	04.03.12.05 (Flächensensor) 04.03.15.08 (Zeilensensor)
Liste mit allen enthaltenen PID-FIDs erstellen.	DFCFingerprintList	Nicht erforderlich	---
Rohdaten einer Timeboy Liste importieren.	DFCMakeTimeboyList	Zunächst nicht geplant. Wird mit späterer Version realisiert.	04.03.xx.xx

Importierte Timeboy Listendaten in Gerät schreiben.	DFCLoadTimeboyList	Zunächst nicht geplant. Wird mit späterer Version realisiert.	04.03.xx.xx
Timeboy Listenspeicher für Import löschen.	DFCClearTimeboyListBuffer	Zunächst nicht geplant. Wird mit späterer Version realisiert.	04.03.xx.xx
Starten des Moduls zur aktiven Verbindungsannahme.	DFCStartActiveConnection	Wird durch http Server erledigt.	---
Stoppen des Moduls zur aktiven Verbindungsannahme.	DFCStopActiveConnection	Wird durch http Server erledigt.	---
Nummer des ersten "Aktiv" verbundenen Kanals ermitteln.	DFCGetFirstActiveChannelID	Nicht erforderlich – das Gerät baut die Verbindung auf.	---
Nummer des nächsten "Aktiv" verbundenen Kanals ermitteln.	DFCGetNextActiveChannelID	Nicht erforderlich – das Gerät baut die Verbindung auf.	---
Informationen zum "Aktiv" verbundenen Kanals abrufen.	DFCGetInfoActiveChannel	Nicht erforderlich – das Gerät baut die Verbindung auf.	---
An/Abschaltung der Annahme von der Meldung "Datensatz verfügbar".	DFCSetRecordAvailable	Nicht erforderlich – das Gerät baut die Verbindung auf.	---
Zugriff auf die Warteschlange der Meldung "Datensatz verfügbar".	DFCRecordAvailable	Nicht erforderlich – das Gerät baut die Verbindung auf.	---
Ein Gerät an diese Kanalnummer binden.	DFCBindDeviceToChannel	Nicht erforderlich – das Gerät baut die Verbindung auf.	---
Setzen welche Blocktypen mit Unterbrechung übertragen werden sollen	DFCBlockTransferSetDuration	Nicht sinnvoll für http Datenaustausch.	---
Wiederaufnehmen einer unterbrochenen Blockübertragung	DFCBlockTransferResume	Nicht sinnvoll für http Datenaustausch.	---
Den aktuellen Typen und Zustand ermitteln	DFCBlockTransferGetState	Nicht sinnvoll für http Datenaustausch.	---
Verwerfen einer bestehender Blockübertragung	DFCBlockTransferDiscard	Nicht sinnvoll für http Datenaustausch.	---

Informationen über Erfolg bzw. Fehler einer per http ausgelösten Aktion erfolgt über Systemmeldungen.

A.2: Vergleich von Datafox Studio und http Level 1

Funktion im Datafox Studio	Flussrichtung	Funktion http Level 1	Status / Plan
Upload der Firmware	-> Gerät	- Web-Server übermittelt Aufforderung zum Firmware-Update an das Gerät - Gerät übermittelt seine Firmware-Version an den Webserver - Webserver prüft, welche Firmware er zum Aktualisieren des Geräts bereitstellen kann - Gerät lädt herunterzuladende Firmware [0xDF01]	04.03.xx.xx
Upload des Setups	-> Gerät	Gerät lädt Setup Datei [0xDF02]	04.03.12.01
Download des Setups	-> Studio	df_send_file=<PATH ON SRV>,setup, Transferdateityp 0xDF0C (2.2.3.2.16)	04.03.12.01

Upload von Setup-Listen	-> Gerät	df_setup_list (2.2.3.2.19) [0xDF03]	04.03.10.01
Upload von ZK-Listen	-> Gerät	df_ac2_list (2.2.3.2.19) [0xDF04]	04.03.10.01
Upload von Timeboy-Listen	-> Gerät	Transferdateityp 0xDF05	04.03.xx.xx
Daten lesen, löschen	-> Studio	Werden aktiv vom Gerät übermittelt	04.03.10.00
Kamerabilder lesen	-> Studio	Gerät sendet neue Bilder per Upload-Form an den Webserver als Transferdatei [0xDF06]. Der Upload-Pfad auf dem Server ist über die Systemvariable COM.HTTP_MODE[n].SEND_IFF festgelegt.	04.03.18.04
Seriennummer lesen	-> Studio	df_kvp=serialnumber (2.2.3.2.12)	04.03.12.01
Uhr stellen	-> Gerät	df_time (2.2.3.2)	04.03.10.00
Nachricht senden	-> Gerät	df_msg (2.2.3.2.4)	04.03.10.01
Globale Variable lesen	-> Studio	df_kvp=var,<VAR_NAME> (2.2.3.2.12)	04.03.12.01
Sprachdatei übertragen	-> Gerät	Transferdateityp 0xDF07	04.03.18.04
Farbdatei TimeboyIV übertragen	-> Gerät	Transferdateityp 0xDF08	04.03.xx.xx
LAN / WLAN Konfiguration übertragen	-> Gerät	Transferdateityp 0xDF09	04.03.xx.xx
Touchkonfigurationsdatei (EVO 4.3) übertragen	-> Gerät	Transferdateityp 0xDF0F	04.03.xx.xx
Displaydesign übertragen	-> Gerät	Transferdateityp 0xDF00 mit Dateiname „mainmenu.bin“	04.03.20.01
U&Z Konfiguration übertragen	-> Gerät	Transferdateityp 0xDF0B	04.03.xx.xx
Systemvariablen lesen	-> Studio	df_kvp=var,<VAR_NAME> (analog zu globalen Variablen) (2.2.3.2.12)	04.03.12.01
Systemvariablen setzen	-> Gerät	df_var (2.2.3.2.2)	04.03.10.01
Gerätelog auslesen	-> Studio	df_send_file=<PATH ON SRV>,syslog, Transferdateityp 0xDF0C (2.2.3.2.16)	04.03.12.01
Update BioKey Modul	-> Gerät	Transferdateityp 0xDF12 bis 0xDF15	04.03.xx.xx
Fingerprint-Templates lesen	-> Studio	df_send_file=<PATH ON SRV>,finger,..., Transferdateityp 0xDF18 (2.2.3.2.16)	04.03.12.05 (Flächensensor) 04.03.15.08 (Zeilensensor)
Fingerprint-Templates löschen	-> Gerät	df_remove_finger=<PID>,<FID> (2.2.3.2.18)	04.03.12.05 (Flächensensor) 04.03.15.08 (Zeilensensor)

Fingerprint-Templates schreiben	-> Gerät	Transferdateityp 0xDF18	04.03.12.05 (Flächensensor) 04.03.15.08 (Zeilensensor)

A.3: Aufbau einer Transferdatei

Dem Aufbau einer vom Server zum Gerät bzw. vom Gerät zum Server übertragenen Datei liegt das IFF-Dateiformat zu Grunde. Dieses Dateiformat erlaubt die Trennung von Daten innerhalb einer Datei in „Chunks“. Der Inhalt eines Chunks kann in Abhängigkeit des Chunk-Typen interpretiert werden, unbekannte Chunk-Typen können beim Interpretieren der IFF-Datei überlesen werden.

Setup- und Zutrittslisten können entweder innerhalb einer IFF-Datei oder direkt per `df_setup_list` und `df_ac2_list` an die Geräte verteilt werden.

Der generelle Aufbau der übrigen, zu übertragenden Dateien in IFF-Form als Liste stellt sich wie folgt dar.

	Offset	Länge	Name	Beschreibung	Beispiel
DFIF-Header	0	4	Header	Der Chunk ist vom Typ „FORM“	„FORM“
	4	4	Datenlänge	Länge der Datei minus 8 (so vielen Daten sind noch zu lesen)	16568
	8	4	Form Typ Kennung	Id der Form	„DFIF“
Versionsdaten	12	4	Header	Der Chunk ist vom Typ „DFFV“	„DFFV“
	16	4	Datenlänge	Gesamtlänge des Forms ohne die 8 Header-Bytes	28
	20	12	FW-Version	Firmware-Versionsstring	„04.03.11.00\0“
	32	16	FW-Suffix	Kennung bei Beta- oder RC-Versionen	„https.1\0“
Datei 1	48	4	Header	Typ des Forms	„FORM“
	52	4	Datenlänge	Länge des Forms (-8)	16392
	56	4	Form Typ Kennung	Kennung für eine zu übertragende Datei	„DFF0“
Dateityp	60	4	Chunk-Header	Zusatzinformationen zur Datei	„FTYP“
	64	4	Datenlänge	Länge des Chunks (-8)	4
	68	2	Dateityp	Kennung des Inhalts gemäß Aufstellung unten	0xDF02 (Setup)
	70	2	CRC (0xA001) über Daten	CRC-Prüfsumme des Inhalts des folgende Data-Chunks (DATA) – ohne Chunk Header	0xF00D
Dateiname	72	4	Chunk-Header	Chunk-Header Dateiname	„FNAM“
	76	4	Datenlänge	Länge des Chunks (-8)	9
	80	9	Dateiname	Name der Datei	„setup.aes“
	89	1	Padding-Byte	Füll-Byte, da die Länge des Dateinamens ungerade ist.	0

	Daten	90	4	Chunk-Header	Beginn des Data-Chunks	„DATA“
		94	4	Datenlänge	Gesamtlänge des Chunks ohne die 8 Header-Bytes	sz
		98	16315	Daten	Inhalt der Datei	
		16413	1	Padding-Byte	Füll-Byte, da sz (16315) ungerade ist.	0
Datei 2	16414		4	Form-Id	Zweite Form mit einer zweiten Datei	„FORM“
	16418		4	Datenlänge	Zweite Form mit einer zweiten Datei Weitere Forms und Chunks der Datei 2	127
	16422		4	Form Typ Kennung		„DFF0“
	16426		123			
	16549		1	Padding-Byte	Füll-Byte, da sz (123) ungerade ist.	0
Signatur	16550		4	Chunk-Header	Chunk-Header Signatur – dieser bezieht sich auf den folgenden Daten-Chunk	„SIGN“
	16554		4	Datenlänge	Länge des Chunks (-8)	18
	16558		2	Algorithmus	Signatur-Algorithmus 0=MD5+AES, 1=SHA1+AES, 27=SHA256+PubKey	0x0001
	16560		16	Signatur-Daten	Signatur über den Inhalt des DFF0-Forms (ohne eventuelles Padding-Byte!)	
Datei 3	16578		4	Form-Id	Dritte, signierte Form mit einer weiteren Datei	„FORM“
	16582		4	Datenlänge	Länge des Forms (-8)	
	16586		4	Form Typ Kennung		„DFF0“
	16590		773	Daten		
	17367		1	Padding-Byte	Füll-Byte, da sz (773) ungerade ist (dieses letzte Padding-Byte ist nicht Bestandteil der signierten Daten)	

Eine Datei kann dabei beliebig viele Chunks enthalten – die Erkennung, dass es sich um eine Datei aus dem Datafox-Kontext handelt, erfolgt am FORM-Typ „DFIF“ zu Beginn der Datei.

Hinweis:
Das IFF-Format ist durch Electronic Arts im Jahr 1985 spezifiziert worden.
Zu dieser Zeit war die Nutzung der Big-Endian-Kodierung für Zahlen im Speicher gebräuchlich. Daher sind die als **Chunk-Längen** genutzten Zahlen auch im **Big-Endian**-Format abzulegen und nicht, wie heute gebräuchlich, im Little-Endian-Format.

Beispiel: Ein Chunk hat die Länge 123456 = 0x1E240 Bytes. In Big-Endian wird dieses durch die Byte-Folge 0x00 0x01 0xE2 0x40 kodiert.

Vgl. https://de.wikipedia.org/wiki/Interchange_File_Format und https://de.wikipedia.org/wiki/Byte-Reihenfolge.
Standard:
http://wiki.amigaos.net/wiki/EA_IFF_85_Standard_for_Interchange_Format_Files

Achtung:
Wenn Sie in Chunks Inhalte einbetten, die eine ungerade Länge habe (etwa Datei-Inhalte, Dateinamen, etc.), ist es erforderlich, dass der Chunk mit einem 0-Byte aufgefüllt wird (padding). Dieses Padding-Byte ist nicht Bestandteil der Länge und wird automatisch erwartet, wenn der nächste Chunk nicht auf einer 2-Byte-Grenze starten würde.

Konkret:
- Ein Chunk belegt immer eine gerade Anzahl von Bytes, selbst wenn der Inhalt des Chunks eine ungerade Länge hat.
- Das Padding-Byte geht **nicht** in eine etwaige CRC-Berechnung ein.

A.3.1: Forms und Chunks innerhalb der Transferdatei

Die Transfer-Datei wird als FORM-Typ übermittelt. Die ID des globalen Form-Headers ist „DFIF“ – wie im vorherigen Kapitel dargestellt. Dem FORM-Header folgt ein Chunk mit Versionsinformation zur Datei („DFFV“), optional ein FORM „DESC“, der den Inhalt der Transferdatei menschenlesbar beschreibt, und eine Liste von Dateien, die als FORMs „DFF0“ kodiert werden.

A.3.1.1: Versions-Informationen [Chunk „DFFV“]

Die Versions-Informationen des Übertragenden werden wie folgt dargestellt:

	Offset	#	Name	Beschreibung	Beispiel
Version	+0	4	Header	Form-ID	„DFFV“
	+4	4	Dateilänge	Gesamtlänge des Chunks ohne die 8 Header-Bytes	28
	+8	12	FW-Version	Firmware-Versionsstring	„04.03.11.00\0“
	+20	16	FW-Suffix	Kennung bei Beta- oder RC-Versionen	„https.1\0“

A.3.1.2: Beschreibung des Dateiinhalts [FORM „DESC“]

IFF-Dateien werden typischer Weise durch das Datafox Studio auf Geräte übertragen. Damit der Anwender weiß, welche Daten innerhalb der IFF Datei enthalten sind, kann diese Informations-FORM eingearbeitet werden.

Im „DESC“ FORM sind Chunks enthalten, die eine hierarchische Einordnung des Dateiinhalts („HIER“) ermöglichen und den Inhalt erklären („HTML“).

	Offset	#	Name	Beschreibung	Beispiel
Beschreibung	+0	4	Header	Form-ID	„FORM“
	+4	4	Form-Länge	Gesamtlänge des Forms ohne die 8 Header-Bytes	4
	+8	4	Form Typ Kennung	Beschreibungs-Form	„DESC“

A.3.1.2.1: Hierarchie-Tag zur Beschreibung [Chunk „HIER“]

Mit einem oder mehrere Hierarchie-Tags kann der Inhalt der Datei organisatorisch einsortiert werden. Mit zwei Tags könnte z.B. ein Update für das WLAN-Modul unter „Geräte-Module“ / „WLAN“ einsortiert werden.

	Offset	#	Name	Beschreibung	Beispiel
Hierarchie	+0	4	Header		„HIER“
	+4	4	Länge	Gesamtlänge des Chunks ohne die 8 Header-Bytes	Sz + 6
	+8	4	Sprache	Sprache, in der das Hierarchie-Tag abgefasst ist (falls es unterschiedliche Sprachen geben wird)	„DE\0“
	+12	2	Ebene	Hierarchie-Ebene, bei 0 beginnend	0
	+14	sz	Text	Beschreibung der Ebene	„Module“ oder „WLAN“
	+14 + sz	0/1	Opt. Padding-Byte	Optionales Füll-Byte, falls sz ungerade ist.	0

A.3.1.2.1: Beschreibungstext [Chunk „HTML“]

Im HTML-Chunk des Beschreibungs-Forms wird eine Beschreibung des Dateiinhalts „in Landessprach“ bereitgestellt. Als Kodierung wird HTML verwendet, wobei hier der Fokus eher auf Inhalt denn auf Stylesheet gelegt werden sollte.

	Offset	#	Name	Beschreibung	Beispiel
B	+0	4	Header		„HTML“

	+4	4	Länge	Gesamtlänge des Chunks ohne die 8 Header-Bytes	Sz + 4
	+8	4	Sprache	Sprache, in der die Beschreibung verfasst ist (falls es unterschiedliche Sprachen geben wird)	„DE\0“
	+12	sz	Text	Beschreibung des Inhalts des Transfer-Datei	
	+12 + sz	0/1	Opt. Padding-Byte	Optionales Füll-Byte, falls sz ungerade ist.	0

A.3.1.3: Übertragen einer Datei [FORM „DFF0“]

Header-Form für eine übermittelte Datei. Innerhalb der FORM sind Chunks enthalten, die

- den Dateinamen,
- das Encoding der Daten [Optional, Default: Binäre Übernahme]
- Zusatzinformationen für die Interpretation [abhängig vom Dateityp]
- Informationen zur (HW-) Kompatibilität
- und den Inhalt der Datei

enthalten.

	Offset	#	Name	Beschreibung	Beispiel
File Header	+0	4	Header		„FORM“
	+4	4	Dateilänge	Gesamtlänge des Chunks ohne die 8 Header-Bytes	4 + Sz
	+8	4	Form Type Kennung	Datafox File Version 0	„DFF0“

Sz ist die Größe der enthaltenen Forms / Chunks.

A.3.1.3.1: Datentyp [CHUNK „FTYP“]

Der „FTYP“ Chunk ist innerhalb einer Transfer-Datei (FORM „DFF0“) enthalten und spezifiziert, wie die Daten der Datei geprüft werden können (CRC) und welche Art Inhalt in der Datei enthalten ist.

	Offset	#	Name	Beschreibung	Beispiel
FTYP Chunk	+0	4	Chunk-Header	Typ-Daten der Datei	„FTYP“
	+4	4	Chunk-Länge	Länge der Zusatzdaten	4
	+8	2	Dateityp	Typ gemäß angehängter Spezifikation	0xDF01
	+10	2	CRC (0xA001) über Daten	Standard-CRC Prüfsumme über Daten, die im DATA oder DATE chunk enthalten sind.	0xF00D

			Init: 0xffff XorOut: 0	Liegen die Daten verschlüsselt vor, so wird der CRC über die verschlüsselten Daten berechnet. Auf diese Weise wird kein Hinweis auf die Verschlüsselung bereitgestellt und die Integrität bleibt auch ohne Schlüssel prüfbar.	

Unser C-Code zur Berechnung der CRC-Prüfsumme über den Datenblock (`data` und `size` beschreiben den Datenblock, `crc` wird beim Aufruf auf `0xffff` gesetzt):

```
// CRC 16bit berechnen
unsigned short GetCRC16(const unsigned char *data, unsigned int size, unsigned short crc)
{
    unsigned int i;
    unsigned char j;

    for ( i = 0; i < size; i++ )
    {
        crc = crc ^ data[i];
        for ( j = 0; j < 8; j++ )
        {
            if ( crc & 0x1 )
            {
                crc >>= 1;
                crc = crc ^ 0xA001;
            }
            else
            {
                crc >>= 1;
            }
        }
    }

    return crc;
}
```

A.3.1.3.2: Zusatzparameter [CHUNK „FAUX“]

Dieser Chunk enthält Zusatzdaten, die für einen Dateitransfer notwendig sein können. Die erforderlichen Zusatzdaten werden im Anhang mit der Definition der Dateitypen vorgegeben. Sind die Zusatzdaten zu einer Datei nicht vorhanden, ist es für das Gerät nicht möglich, die Daten zu verarbeiten – die Datei wird entsprechend abgelehnt:

	Offset	#	Name	Beschreibung	Beispiel
Aux	+0	4	Header		„FAUX“
	+4	4	Länge	Gesamtlänge des Chunks ohne die 8 Header-Bytes	sz
	+8	2	Version	Versionsnummer des Aux-Chunks – bezogen auf den umgebenden Dateitype	1
	+10	sz	Auxdata	Zusatzdaten zur Interpretation des Chunks	z.B. PID und FID eines Fingertemplates

	+10 + sz	0/1	Opt. Padding-Byte	Optionales Füll-Byte, falls sz ungerade ist.	0

Anmerkung: Die Version im FAUX-Chunk startet bei 1. Zum Inhalt des FAUX Chunks sehen Sie bitte auch im Anhang A.3.2 nach.

A.3.1.3.3: Dateiname [CHUNK „FNAM“]

Chunk mit dem Dateinamen:

	Offset	#	Name	Beschreibung	Beispiel
Filename	+0	4	Header		„FNAM“
	+4	4	Länge	Gesamtlänge des Chunks ohne die 8 Header-Bytes	sz
	+8	sz	Dateiname	Name der Datei	„my.cert“
	+8 + sz	0/1	Opt. Padding-Byte	Optionales Füll-Byte, falls sz ungerade ist.	0

A.3.1.3.4: Encoding-Informationen des Datenblocks [CHUNK „ENC “]

Chunk mit Encoding-Informationen für den Dateiinhalt:

	Offset	#	Name	Beschreibung	Beispiel
Zeichen-Encoding	+0	4	Header	Header für das Encoding	„ENC “
	+4	4	Länge	Gesamtlänge des Chunks ohne die 8 Header-Bytes	sz
	+8	sz	Encoding	Mime-Type für das Encoding, in dem der Datenblock (nach vorheriger, optionaler Entschlüsselung) interpretiert werden muss. Fehlt dieses Feld, werden die Daten als Binär-Daten interpretiert.	ISO-8859-1
	+8 + sz	0/1	Opt. Padding-Byte	Optionales Füll-Byte, falls sz ungerade ist.	0

A.3.1.3.5: Kompatibilitäts-Informationen [CHUNK „COMP“]

Bei Update-Daten, z.B. für auf dem Gerät eingesetzte Module, enthält dieser Chunk Informationen zur Kompatibilität. Der Inhalt der internen Struktur wird hier nicht definiert.

	Offset	#	Name	Beschreibung	Beispiel
K	+0	4	Header		„COMP“

	+4	4	Länge	Gesamtlänge des Chunks ohne die 8 Header-Bytes	2 + sz
	+8	2	Version	Version des Kompatibilitäts-Chunks	1
	+10	sz	Data	Kompatibilitätsdaten	
	+10 + sz	0/1	Opt. Padding-Byte	Optionales Füll-Byte, falls sz ungerade ist.	0

A.3.1.3.6: Datei-Inhalt [CHUNK „DATA“]

Hinweis: Es gibt alternativ zum DATA-Chunk „intern verschlüsselte“ Chunks, die z.B. für Firmware-Updates eingesetzt werden (siehe Chunk „DATE“)

Chunk mit dem Dateiinhalt:

	Offset	#	Name	Beschreibung	Beispiel
Filedata	+0	4	Header		„DATA“
	+4	4	Dateilänge	Gesamtlänge des Chunks ohne die 8 Header-Bytes	sz
	+8	sz	Daten	Inhalt der Datei	
	+8 + sz	0/1	Opt. Padding-Byte	Optionales Füll-Byte, falls sz ungerade ist.	0

A.3.1.3.7: Aufbau eines intern verschlüsselten Dateielements (alternativ zum DATA Chunk) [CHUNK „DATE“]

Innerhalb einer Datei vom Typ 0xDF01 (Firmware Update) können mehrere der in diesem Kapitel beschriebenen Data-Chunks enthalten sein, um Text, Programmflash, etc. zu übermitteln. Diese Chunks sollten nicht außerhalb von Datafox erstellt werden.

	Offset	#	Name	Beschreibung	Beispiel
Verschlüsseltes Datensegment	+0	4	Header	Chunk für verschlüsselte Daten	„DATE“
	+4	4	Dateilänge	Gesamtlänge des Chunks ohne die 8 Header-Bytes	20 + sz
	+8	2	Version	Versionsnummer dieses Firmware-Update-Chunks	
	+10	2	Internal Type	Interner Dateityp (Flash, Fonts, Texte, Icons, …)	
	+12	4	Start-Offset	Start-Offset des Firmware-Updates im Geräte-CPU-Flash	0x00208000
	+16	4	End-Offset	End-Offset des Firmware-Updates im Geräte-CPU-Flash	0x003fffff
	+20	4	CRC	CRC über die entschlüsselten Daten	

	+24	4	Seed	Seed-Value für die Entschlüsselung	0x12345678
	+28	sz	Daten	Inhalt der Datei	

A.3.1.3.8: Signatur-Chunk [CHUNK „SIGN“]

Der Signatur-Chunk bezieht sich auf den folgenden Chunk in der IFF-Datei.

	Offset	#	Name	Beschreibung	Beispiel
Signatur	+0	4	Header	Chunk für signierte Daten	„SIGN“
	+4	4	Dateilänge	Gesamtlänge des Chunks ohne die 8 Header-Bytes	2 + sz
	+8	2	Signaturmethode	0 – MD5	0
	+10	sz	Signatur-Daten		

A.3.1.3.9: Aufbau eines signierten Daten-Segments [CHUNK „DATS“]

Innerhalb einer Datei vom Typ 0xDF01 (Firmware Update) können mehrere der in diesem Kapitel beschriebenen Data-Chunks enthalten sein, um Text, Programmflash, etc. zu übermitteln. Diese Chunks sollten nicht außerhalb von Datafox erstellt werden – daher ist für die Verarbeitung durch die Geräte-Firmware ein Signatur-Chunk vom Typ „SIGN“ erforderlich.

	Offset	#	Name	Beschreibung	Beispiel
Datensegment	+0	4	Header	Chunk für signierten Daten	„DATS“
	+4	4	Dateilänge	Gesamtlänge des Chunks ohne die 8 Header-Bytes	12 + sz
	+8	2	Version	Versionsnummer dieses Firmware-Update-Chunks	
	+10	2	Internal Type	Interner Dateityp (Flash, Fonts, Texte, Icons, …)	
	+12	4	Start-Offset	Start-Offset des Firmware-Updates im Geräte-CPU-Flash	0x00208000
	+16	4	End-Offset	End-Offset des Firmware-Updates im Geräte-CPU-Flash	0x003fffff
	+20	sz	Binär-Daten	Inhalt der Datei	

A.3.1.4: Datensatz- / Listen-Beschreibung [FORM „DFDS“]

Header-Form für eine Datensatz- oder Liste-Beschreibung. Innerhalb der Form sind folgende Chunks enthalten:

- Name der Datensatzbeschreibung
- Index der Datensatzbeschreibung im Setup
- Index des Prioritätenfeldes (optional)

- Index des Schlüsselfeldes (optional)

Des Weiteren können DCOL-Forms enthalten sein, die die einzelnen Felder des Datensatzes beschreiben (Vgl. A.3.1.5).

	Offset	#	Name	Beschreibung	Beispiel
Dataset	+0	4	Header		„FORM“
	+4	4	Länge	Gesamtlänge des FORMs ohne die 8 Header-Bytes	4 + Sz
	+8	4	Form Type ID	Datafox Data Structure	„DFDS“

A.3.1.4.1: Name der Datensatzbeschreibung [„DNAM“]

Chunk mit dem Namen der Datensatzbeschreibung. Der Name enthält den Typ des Datensatzes und den im Setup vergebenen Namen getrennt durch einen Punkt. Die Setup-Liste mit den Namen „Personal“ wird folglich als „list.personal“ repräsentiert.

Folgenden Präfixe werden eingesetzt:

- „list“ gibt an das es eine im Setup definierte Liste ist,
- „access“ steht für eine Liste der Zutrittskontrolle,
- „record“ steht für eine Datensatz-Beschreibung.

	Offset	#	Name	Beschreibung	Beispiel
Datasetnam	+0	4	Header		„DNAM“
	+4	4	Länge	Gesamtlänge des Chunks ohne die 8 Header-Bytes	sz
	+8	sz	Dateiname	Name des Datensatzes	„record.Buchungen“
	+8 + sz	0/1	Opt. Padding-Byte	Optionales Füll-Byte, falls sz ungerade ist.	0

A.3.1.4.2: Index der Datensatzbeschreibung im Setup [„DIDX“]

Der Chunk beinhaltet die Indexnummer der Datensatzbeschreibung im Setup. Diese startet mit 0 und wird hoch gezählt.

	Offset	#	Name	Beschreibung	Beispiel
Dataset Index	+0	4	Header		„DIDX“
	+4	4	Länge	Gesamtlänge des Chunks ohne die 8 Header-Bytes	2
	+8	2	Index	Index der Datenbeschreibung im Setup	1

A.3.1.4.3: Index des Prioritätfeldes [„DPRI“]

Dieser Chunk beinhaltet einen den Index des Feldes, welches die Informationen für die Priorität des Datensatzes enthält. Falls kein Feld für die Priorität ausgewählt wurde fehlt dieser Chunk.

	Offset	#	Name	Beschreibung	Beispiel
Prio - Index	+0	4	Header		„DPRI“
	+4	4	Länge	Gesamtlänge des Chunks ohne die 8 Header-Bytes	2
	+8	2	Index	Index des Feldes, welches Prioritätsinformationen enthält	1

A.3.1.4.4: Index des Schlüsselfeldes [„DKEY“]

Dieser Chunk enthält den Index des Schlüsselfeldes, falls dieser gesetzt wurde.

	Offset	#	Name	Beschreibung	Beispiel
Schlüssel	+0	4	Header		„DKEY“
	+4	4	Länge	Gesamtlänge des Chunks ohne die 8 Header-Bytes	2
	+8	2	Index	Index des Schlüsselfeldes	1

A.3.1.5: Spalteninformation für Listen und Datensätze [FORM „DCOL“]

Header-Form für die Beschreibung eines Feldes einer Datensatz- oder Listenbeschreibung. In der Form sind folgende Chunks enthalten:

- Name des Feldes
- Informationen zum Feldaufbau

	Offset	#	Name	Beschreibung	Beispiel
Column	+0	4	Header		„FORM“
	+4	4	Dateilänge	Gesamtlänge des FORMs ohne die 8 Header-Bytes	4 + Sz
	+8	4	Form Type Kennung	Datafox Column Data	„DCOL“

A.3.1.5.1: Informationen zum Feldaufbau [„CINF“]

Dieser Chunk enthält den Datentyp des Feldes.

	Offset	#	Name	Beschreibung	Beispiel
F	+0	4	Header		„CINF“

	+4	4	Länge	Gesamtlänge des Chunks ohne die 8 Header-Bytes	6
	+8	2	Typ	Typ des Feldes als Zahlencode: 2 – Datum und Uhrzeit 3 – Numerischer Wert 4 – Alphanumerischer Wert 7 – Fingerprint Template DIN V44600 (161 Byte) 8 – Fingerprint Template Idencom Compact (216 Byte) 9 – Binärdaten (max. 220 Byte)	4
	+10	2	Größe	Größe des Feldes in Byte	35
	+12	2	Index	Index des Feldes	5

A.3.1.5.2: Namen des Feldes [„CNAM“]

In dem Feld wird der Name des Feldes gespeichert.

	Offset	#	Name	Beschreibung	Beispiel
Feldname	+0	4	Header		„CNAM“
	+4	4	Länge	Gesamtlänge des Chunks ohne die 8 Header-Bytes	sz
	+8	sz	Feldname	Name des Feldes als String	„Personalnummer“
	+8 + sz	0/1	Opt. Padding-Byte	Optionales Füll-Byte, falls sz ungerade ist.	0

A.3.2: Dateitypen

Jede Datei, die zwischen Gerät und Server ausgetauscht wird, enthält einen Dateitypen. Dieser regelt die Interpretation der Datei – die ersten beiden Bytes der „Version“ entsprechen der Version aus Abschnitt A.3.1.3.2: Zusatzparameter [CHUNK „FAUX“], daher sind diese in der folgenden Tabelle kursiv dargestellt.

Dateityp	Funktion der Datei	Gerät -> Server	Server -> Gerät
0xDF00	Datei bezogen auf das Filesystem des Gerätes. FNAM-Inhalt: Kompletter Pfad im Geräte-Dateisystem, falls die Datei nicht im obersten Verzeichnis auf dem Gerät abgelegt ist. (Vgl.2.2.3.2.16)		
0xDF01	Firmware Datei *	NEIN	

0xDF02	Setup Datei		
0xDF03	Setup-Listendatei Aux-Parameter: [*Version >> 8, Version,* Listenname]		
0xDF04	ZK2-Listendatei Aux-Parameter: [*Version >> 8, Version,* Listenname]		
0xDF05	Listendatei Timeboy (wird vom MasterIV-Gerät dann zum Timeboy übertragen). Aux-Parameter: [*Version >> 8, Version,* GroupID >> 8, GroupID, Listenname]	NEIN	
0xDF06	Kamera-Bild, Unterschrift oder Barcode (vom Gerät erfasste Dateien)		NEIN
0xDF07	Sprachdatei		
0xDF08	Farbdatei Timeboy		
0xDF09	LAN / WLAN Konfigurationsdatei		
~~0xDF0A~~	~~Displaydesign-Datei~~ Bitte nutzen Sie 0xDF00 mit Dateiname „mainmenu.bin“		
0xDF0B	U&Z Konfigurationsdatei		
0xDF0C	Systemlog		NEIN
0xDF0D	Bootloader *	NEIN	
0xDF0E	Fingerprint-Daten (nur Saturn 01) Aux-Parameter: [*Version >> 8, Version,* PID >> 24, PID >> 16, PID >> 8, PID, GID >> 8, GID, FID] Nutzen Sie bitte 1 als GID (Gruppen-ID). 0 ist ungültig, Werte > 1 werden für Verifikation gegenüber Ausweisen eingesetzt.	NEIN -> 0xDF18	
0xDF0F	Touchkey Konfigurationsdatei		
0xDF10	Hardware-Info-Datei (HIP) *		
0xDF11	Update-Datei für WLAN Modul RS9110 *	NEIN	
0xDF12	Update-Datei für Fingerprint-Sensor Biokey 3000-Modul *	NEIN	
0xDF13	Update-Datei für Fingerprint-Sensor Biokey 4000-Modul *	NEIN	
0xDF14	Update-Datei für Fingerprint-Sensor Biokey 4020-Modul *	NEIN	
0xDF15	Update-Datei für Fingerprint-Sensor Saturn 01 *	NEIN	
0xDF16	Update-Datei für das U&Z Funk-Basisstation (FSM) *	NEIN	
0xDF17	Update-Datei für den Näherungssensor	NEIN	
0xDF18	Fingerprint-Daten (Saturn 01 oder Idencom) Aux-Parameter: [Version >> 8, Version, Template-Typ, PID >> 24, PID >> 16, PID >> 8, PID, GID >> 8, GID, FID]		

	Template-Type: 0 – DIN V66400 Format (161 Byte) 1 – Idencom-Compact Format (216 Byte) 2 – Idencom-Standard Format (561 Byte) 3 – Saturn 01 Binär-Template-Format Nutzen Sie bitte 1 als GID (Gruppen-ID). 0 ist ungültig, Werte > 1 werden für Verifikation gegenüber Ausweisen eingesetzt.		
0xDF19	Verzeichnisinhalt, siehe df_send_file (2.2.3.2.16) mit Parameter „dir“.		NEIN
0xDF1A	Konfigurationsdatenpaket für den TWN4-Lesers, der das lesen unterschiedlicher Transponder ermöglicht (BIX / AppBlaster)	NEIN	
TBA	Update-Datei (Update für ZK-Bus Teilnehmer, …) * Aux-Parameter: [*Version >> 8, Version,* Modul-ID >> 8, Modul-ID]		

* **Hinweis:** Dateien dieses Typs können nur durch Datafox erzeugt werden. Der Inhalt des DATE-Chunks ist mit einem geheimen Datafox-Schlüssel chiffriert, der bei der Auswertung des Gerätes die Authentizität und Integrität des Programmpakets sicherstellt.

Anhang B: https Kommunikation

Die in diesem Handbuch beschriebene http-Kommunikation kann ab Firmware Stand 04.03.11.01 auch verschlüsselt über https erfolgen.

B.1: Elemente der https Infrastruktur

Wie auch http ist https ein Client-Server-Protokoll. Der Client baut eine Verbindung zum Port des https-Servers über TCP/IP auf, der Datenstrom wird zur Absicherung gegen Mithörer verschlüsselt.

Zum Einsatz kommen hierbei sowohl asymmetrische Verschlüsselung (Aushandlung der Verbindung) in Form des Server-Zertifikats wie auch symmetrische Verschlüsselung für den (späteren) Datenaustausch.

B.2: Verbindungsaufbau

Der Ablauf der https Kommunikation erfolgt analog zu der in einem aktuellen Webbrowser umgesetzten Vorgehensweise.

- Aufbau der Verbindung vom Terminal zum https Server.
- Handshake zwischen Client und Server
    - Aushandeln des später einzusetzenden Verschlüsselungsverfahrens
- Austausch der Zertifikate
    - Prüfung des Server-Zertifikats. Ist dieses gültig, so steht dem Client der öffentliche Schlüssel des Servers zur Verfügung. Mit diesem kann der Client Nachrichten an den Server übermitteln, die nur mit dem privaten Schlüssel des Servers entschlüsselt werden können.
    - Optional: Wenn der Server vom Client ein Zertifikat anfragt, so wird der Client dieses bereitstellen, sofern dieses dort hinterlegt ist.
- Austausch der Schlüssel
    - Dieser Schlüssel wird für die symmetrische Verschlüsselung der Kommunikation zwischen Server und Client genutzt.

B.3: Prüfung des Server-Zertifikats

Die Firmware erfordert, dass das vom Server übermittelte Zertifikat bei der https Aushandlung gültig ist. Die Gültigkeit wird dabei „klassisch“ festgestellt, d.h. ein Zertifikat wird gegen auf dem Terminal hinterlegte Referenz-Zertifikate geprüft (das sog. CA-Bundle, das es heute bei jedem Betriebssystem und/oder Webbrowser gibt). Das CA-Bundle definiert hierbei diejenigen Zertifikate, die als per-se gültig akzeptiert werden.

Anwendungsbeispiele:

- Der Server („your-company.de“) hat ein mit einem Verisign-Zertifikat signiertes Zertifikat:

  In diesem Fall übermittelt der Server sein Zertifikat zum Terminal.
  Das Terminal stellt fest, dass das Server-Zertifikat „your-company.de“ mit einem Verisign-Zertifikat erstellt wurde. Es sucht nach diesem Zertifikat im lokalen CA-Bundle, um die Abstammung des „your-company.de“-Zertifikats zu prüfen.

  Folgende Möglichkeiten:

    - Der Client findet das Verisign-Zertifikat nicht: Das Zertifikat „your-company.de“ wird als **ungültig** eingestuft.

- Das übermittelte Zertifikat ist **nicht** vom Verisign-Zertifikat abgeleitet: Das Zertifikat „your-company.de“ wird als **ungültig** eingestuft.
- Das übermittelte Zertifikat ist vom Verisign-Zertifikat abgeleitet: Das Zertifikat „your-company.de“ wird als **gültig** eingestuft.

- Der Server („your-company.de“) hat ein selbst-signiertes Zertifikat:

  In diesem Fall kann die Zertifikatsprüfung nur erfolgreich sein, wenn das Server-Zertifikat im Gerät als vertrauenswürdig hinterlegt wurde.

- Der Server („your-company.de“) hat ein von Datafox signiertes Zertifikat:

  Anders als im ersten Fall übermittelt der Server seine Zertifikatskette. Da Datafox ein Globalsign-signiertes Zertifikat bereitgestellt hat, wird folglich zusätzlich zum Server-Zertifikat das Datafox-Zertifikat (aber nicht das Globalsign-Zertifikat) übermittelt.

  Der Client prüft nun, ob das Server-Zertifikat wirklich vom Datafox-Zertifikat abstammt. Ist dem nicht so, wird das Zertifikat als **ungültig** eingestuft.

  Stimmt die Abstammung, so wird als nächstes geprüft, ob das im Datafox-Zertifikat referenzierte Globalsign-Zertifikat im CA-Bundle auf dem Gerät hinterlegt ist. Ist dieses nicht der Fall, wird das Zertifikat als **ungültig** eingestuft.

  Liegt das Globalsign-Zertifikat auf dem Gerät als vertrauenswürdiges Zertifikat vor, entscheidet die Abstammungsprüfung des Datafox-Zertifikats darüber, ob das Server-Zertifikat **gültig** oder **ungültig** ist.

B.4: Die Kommunikation

Nach erfolgreicher Prüfung des Server-Zertifikats handeln die Kommunikationspartner den zu nutzenden Schlüssel aus („Session Key“). Die Kommunikation erfolgt nun abgesichert durch symmetrische Verschlüsselung. Zum Austausch des Schlüssels wird der öffentliche Schlüssel des Server-Zertifikats durch den Client genutzt, so dass der Server mit seinem privaten Schlüssel die Nachricht des Client entschlüsseln kann.

Die Firmware lehnt die Nutzung der als nicht mehr zeitgemäß (da unsicher) eingestuften Verschlüsselungsverfahren nach Spezifikation TLS 1.0 ab. Es werden lediglich Verfahren akzeptiert, die ab TLS 1.1 eingeführt wurden.

B.5: Nutzung eines selbst-signierten (Server-) Zertifikats

Mittels der OpenSSL-Implementierung (normaler Weise unter Linux bzw. mit Cygwin unter Windows verfügbar) können Sie den privaten und öffentlichen Schlüssel für die https-Kommunikation selbst erstellen. Der private Schlüssel (my.key) wird dabei nur auf dem Webserver hinterlegt, der öffentliche (my.cert) wird vom Server wie auch vom Client benötigt.

Ein neues 2048-bit RSA Schlüsselpaar können Sie mit folgendem Befehl erstellen:

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout my.key -out my.cert
```

Wünschen Sie den Schutz des Key-Files mit einem Password, so können Sie diesen per zusätzlichen Parameter “-passout file:key.txt“ im openssl-Aufruf integrieren lassen. Der Schlüssel muss dazu in der Datei „key.txt“ auf dem Rechner vorliegen, der das openssl-Programm ausführt.

Sie können auch andere Schlüsselpaar erzeugen, etwa einen 3072 Bit langen RSA Schlüssel durch:

```
openssl req -x509 -nodes -days 365 -newkey rsa:3072 -keyout my.key -out my.cert
```

oder ein Schlüsselpaar, das elliptische Kurven (ECC) nutzt:

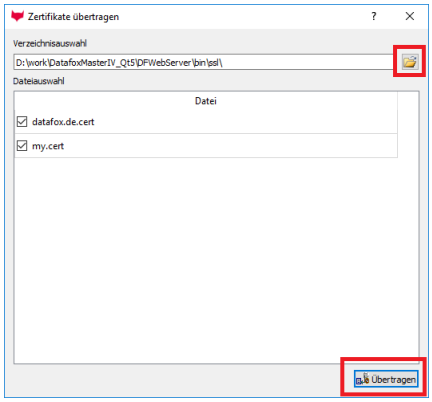
```
openssl ecparam -genkey -name prime256v1 -out key.pem
openssl req -new -sha256 -key key.pem -out csr.csr
openssl req -x509 -sha256 -days 365 -key key.pem -in csr.csr -out certificate.pem
```

Der Prozess zum Erzeugen eines ECC-Schlüsselpaares nutzt einen CSR (Certificate Signing Request), so wie dieser normalerweise mit einer Certificate Authority durchgeführt wird – signiert aber im Anschluss selbst. Zur Nuztung auf dem Gerät müssen dann die erzeugten Dateien noch umbenannt werden, etwa key.pem -> ecc.key und certificate.pem -> ecc.cert bevor diese auf ein Gerät übermittelt werden.

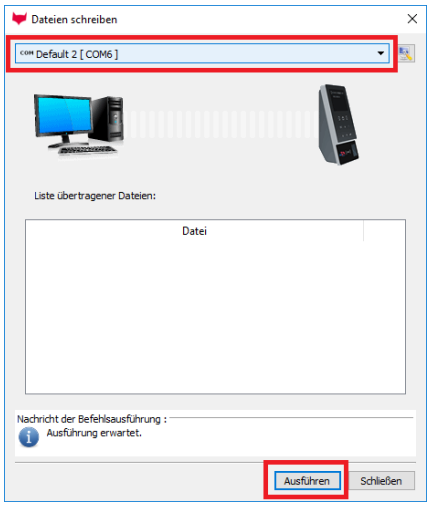
B.5.1: Einrichtung des Terminals – Hinterlegen von Server-Zertifikaten

Wenn Sie mit einem https Server kommunizieren, ist es erforderlich, dass das Terminal die Zertifikatskette des Servers prüfen kann. Dazu muss das Zertifikat, von dem die Zertifikatskette des Serves abgeleitet ist bzw. das selbst-signierte Zertifikat des Servers auf dem Terminal hinterlegt werden.

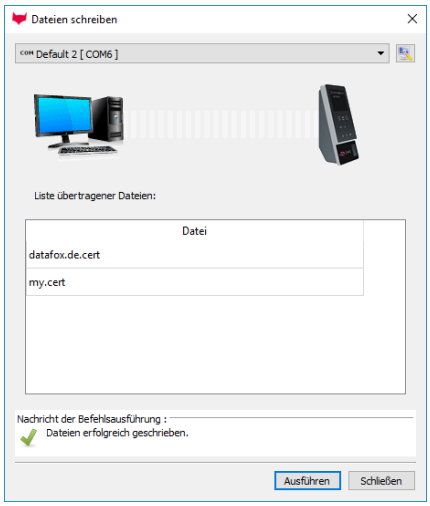
Sie können die Zertifikate auf den Geräten mit dem Datafox Studio ab Version 04.03.11.02 verwalten. Mittels von „Konfiguration“ -> „Zertifikate übertragen“ können Sie gewisse Zertifikate wie folgt auf einem Gerät als vertrauenswürdig hinterlegen.



Auswahl des Verzeichnisses, in dem die *.cert-Dateien liegen. Diese können einzeln für die Übertragung zu- oder abgewählt werden. Durch betätigen von „Übertragen“ wird der nächsten Dialog dargestellt.



Hier können sie das Zielgerät auswählen. Erst bei Betätigung von „Ausführen“ startet der Übertragungsprozess.



Nach erfolgreicher Übertragung werden die auf das Gerät übermittelten Zertifikate aufgelistet.

Achtung:
Bitte übermitteln Sie zu Beginn Ihrer Integrationsarbeiten **nur die notwendigen Zertifikate** (und niemals Keys, diese gehören zum Server). Der Zertifikatsspeicher auf

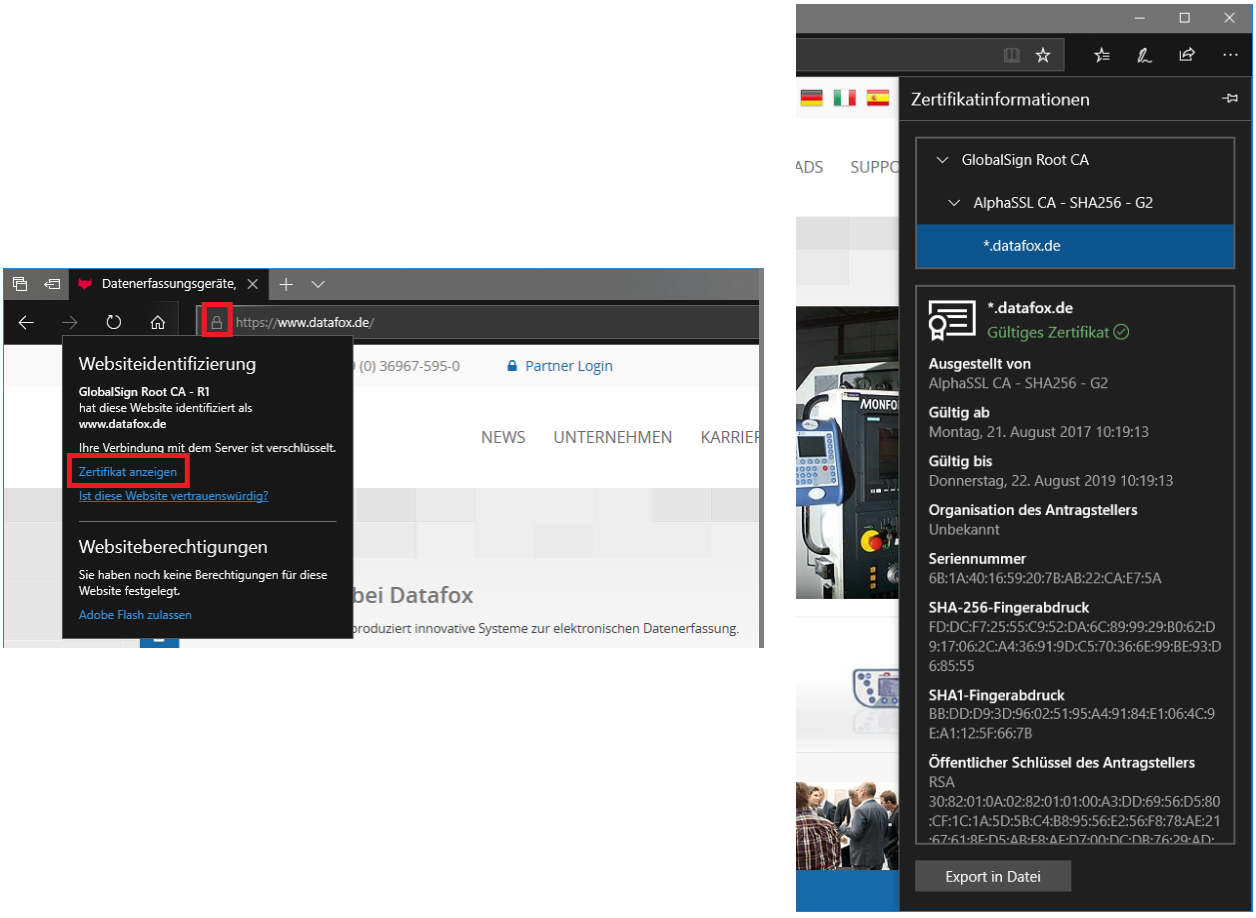
den Terminals ist begrenzt und deutlich kleiner als z.B. in aktuell verfügbaren Web-Browsern oder Betriebssystemen.

Achtung:
Die aktuell verfügbare Version der Firmware liest den Zertifikatsspeicher nur beim Gerätestart ein. Sobald Sie **Zertifikate** auf dem Gerät **ändern**, müssen Sie daher das **Gerät neu starten**!

B.5.2: Welches Zertifikat nutzt mein Web-Server? („Alter“ Edge-Browser)

Falls unklar ist, welches Zertifikat Ihre Webserver einsetzt, können Sie dieses mit aktuellen Browsern wir folgt ermitteln:

Betätigen Sie das Schloss im Microsoft Edge Browser. Es erscheint das Popup „Webseitenidentifizierung“. Dort betätigen Sie den Link „Zertifikat anzeigen“. Die Zertifikatskette der Webseite wird im rechten Bereich des Web-Browsers dargestellt:



Sie erkennen auf der rechten Seite, dass für die Webseite „datafox.de“ ein Wildcard-Zertifikat angewendet wird. Dieses ist vom „AlphaSSL CA“ abgeleitet, welches vom „GlobalSign Root CA“ Zertifikat abstammt.

Da die Zertifikatsprüfung des Server-Zertifikats in TLS gegen vertrauenswürdige Root-Zertifikat erfolgt, benötigt das Terminal zur Kommunikation das „GlobalSign Root CA“-Zertifikat zur Prüfung. Klicken Sie auf dieses und anschließend auf „Export in Datei“. Legen Sie das Zertifikat als „GlobalSignRootCA.crt“ im Dateisystem ab.

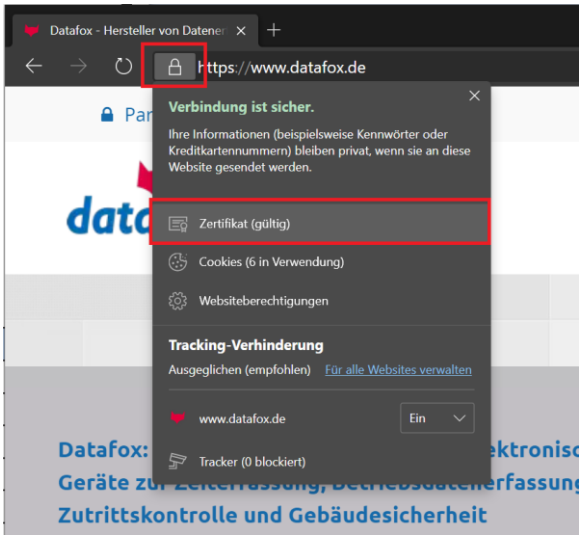
Da das Terminal PEM-kodierte Zertifikate benötigt, das gespeicherte Zertifikat allerdings im Binär-Format (DER) abgelegt ist, muss es für den Einsatz im Terminal konvertiert werden. Dazu können Sie openssl wie folgt einsetzen:

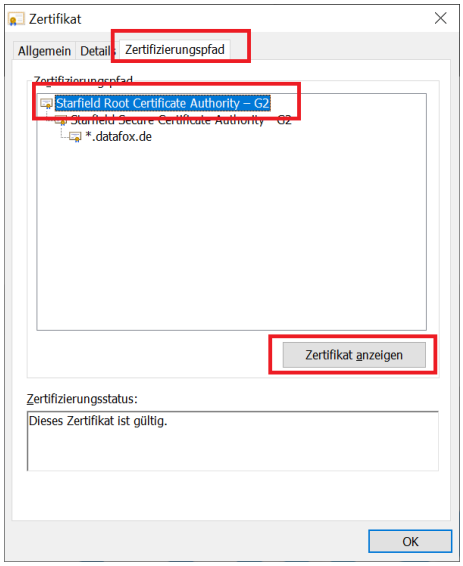
```
openssl x509 -inform DER -in GlobalSignRootCA.crt -out GlobalSignRootCA.pem -text
```

Das entstehende GlobalSignRootCA.pem muss dann noch in eine „GlobalSignRootCA.cert“ umbenannt werden und kann mit dem DatafoxStudioIV auf Ihr Gerät übertragen werden.

B.5.3: Welches Zertifikat nutzt mein Web-Server? („Chromium“ Edge-Browser)

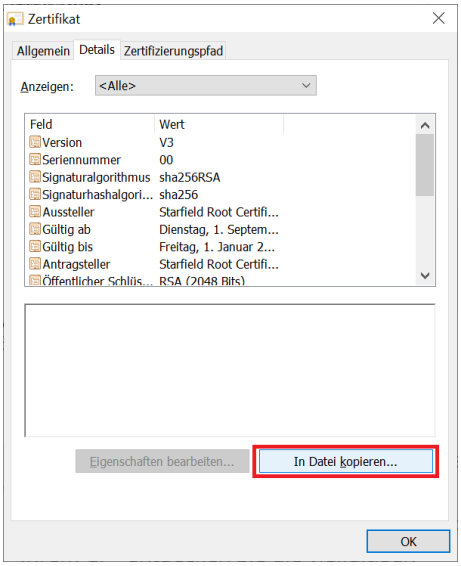
Auch im Chromium-Edge-Browser sind die Zertifikats-Informationen einer Webseite ebenfalls über das Schloss-Symbol zugänglich.

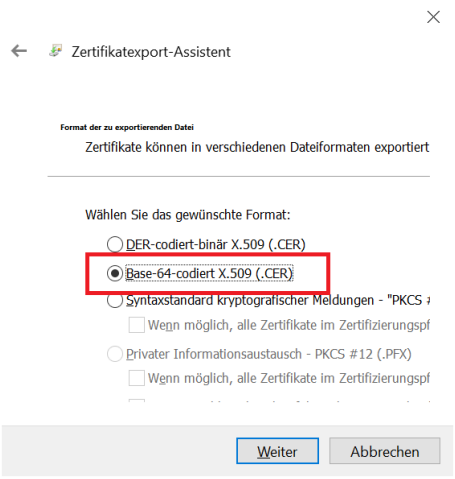




Im folgenden Zertifikatsdialog wechseln Sie bitte auf den Zertifizierungspfad-Reiter, wählen das Root-Zertifikat aus und betätigen die Schaltfläche zum Darstellen des Zertifikats.

Derselbe Dialog erscheint – allerdings für nur noch für das Root-Zertifikat. Wechseln Sie in diesem Dialog auf den Reiter „Details“ und betätigen die Schaltfläche „in Datei kopieren“:





Im folgenden Assistenten zum Zertifikatsexport stellen Sie bitte sicher, dass der Export des Root-Zertifikats als „Base-64-cdiertes X.509“ Zertifikat erfolgt. Das so exportierte Zertifikat ist auf Ihrem Datafox-Gerät nutzbar.

Das Zertifikat hat dann eine Endung “.cer“.
Bennen Sie diese um in „.cert“ und übertagen Sie diese dann an das Gerät.

B.5.4: Einrichtung des Terminals – Hinterlegen von Client-Zertifikaten

Erstellen Sie zunächst ein Schlüssel-Paar für das Gerät. Hierbei kann es sich um ein selbst-signiertes oder ein von einem CA-Zertifikat abgeleitetes Zertifikat handeln.

Übertragen Sie sowohl das Zertifikat als auch den zugehörigen Schlüssel zum Gerät. Wichtig ist hierbei, dass das Zertifikat „client.cert“ und der Schlüssel „client.key“ heißt. Sobald beide Dateien auf dem Gerät vorliegen, erfolgt der TLS Handshake mit Übermittlung des Client-Zertifikats – sofern der Server dieses anfragt.

> **!** **Achtung:**
> Der „client.key“ stellt ein besonders schützenswertes Element im Gerät dar. Wir stellen daher über die Firmware sicher, dass dieser nicht ausgelesen werden kann.
> Sollten Sie folglich den „client.key“ mit dem Datafox Studio auf ein Gerät übertragen haben, so können Sie diesen **nicht** zurücklesen.

B.6: Erstellen einer eigenen CA

OpenSSL ist ein sehr vielseitiges Krypto-Werkzeug und kann nicht nur selbst-signierte Zertifikate erstellen, sondern auch Ketten von Zertifikaten generieren und signieren. Damit wird es möglich, mehrere Zertifikate vom selben Root-Zertifikat abzuleiten – da das Root-Zertifikat für die Zertifikatsprüfung genüg, können Sie auf diese Weise eine Familie von Server- oder Client-Zertifikaten erzeugen.

Installieren Sie bitte zunächst OpenSSL auf dem System, auf dem Sie Ihre CA erzeugen wollen. Hier sind Windows- oder Linux-System gleichermaßen geeignet, auf Mac OS X haben wir diesen Prozess nicht ausprobiert.

B.6.1: Erstellen des Root-Schlüssels/Zertifikats der CA

Gehen Sie dazu wie folgt vor:

- Erzeugen Sie zunächst den Root-Schlüssel „`ca.key`“. Setzen Sie Ihre Daten dabei als „Subject“ ein:
  ```
  openssl req -subj "/C=<Country>/ST=<State>/L=<Stadt>/O=<Organisation>
  /OU=<Orga-Unit>/CN=<Common Name>" -new -newkey rsa:2048 -nodes -out ca.csr
  -keyout ca.key
  ```

- Erstellen Sie nun das passende Zertifikat, das dann die maximale Gültigkeits-Periode für alle abgeleiteten Zertifikat vorgibt (7300 Tage sind etwa 20 Jahre):
  ```
  openssl x509 -signkey ca.key -days 7300 -req -in ca.csr -out ca.pem
  ```

- Falls nicht explizit anders konfiguriert, legt OpenSSL eine „demoCA“ an, in der das Root-Zertifikat erwartet wird und abgeleitete Zertifikat hinterlegt werden. Diese hat folgenden Aufbau:

  ```
  demoCA/
  demoCA/cacert.pem
  demoCA/index.txt
  ```

```
demoCA/newcerts/
demoCA/private/
demoCA/private/cakey.pem
demoCA/serial
```

Legen Sie die o.g. Verzeichnisse an, die Datei index.txt sollte leer sein, serial den Wert „01“ enthalten.

- Kopieren Sie das erzeugte Root-Schlüssel-Paar in die demoCA:

```
cp ca.key ./demoCA/private/cakey.pem
cp ca.pem ./demoCA/cacert.pem
cp ca.pem ca.cert
```

Damit ist Ihre CA einsatzbereit – den cakey.pem im private-Verzeichnis müssen Sie schützen! Mit diesem können dann neue Schlüssel erzeugt werden, das `cacert.pem` wir später benötigt, um die Ableitungskette zu prüfen.

Achtung:
Sollten Sie nun den Schlüssel verlieren, kommt dieses letztendlich dem Verlust der vollständigen CA gleich.

B.6.2: Erstellen von abgeleiteten Schlüsselpaaren

Dadurch, dass die `demoCA` jetzt initialisiert ist, sind zum Erstellen eines abgeleiteten Zertifikats „nur“ noch drei Schritte erforderlich:

- Erstellen eines neuen Schlüssels
  ```
  openssl genrsa -out derived-a.key 2048
  ```
- Erstellen eines Zertifizierungs-Requests
  ```
  openssl req -subj "/C=<Country>/ST=<State>/L=<Stadt>/O=<Organisation>
  /OU=<Orga-Unit>/CN=<Common Name>" -new -key derived-a.key -out derived-
  a.csr
  ```
- Erzeugen des signierten Zertifikats (730 Tage => 2 Jahre Gültigkeit)
  ```
  openssl ca -in derived-a.csr -days 730 -out derived-a.cer
  ```

Damit liegt nun mit `derived-a.key` und `derived-a.cer` ein Schlüsselpaar vor, das von `cakey.pem` abgeleitet ist. Zur Prüfung der Korrektheit der Ableitungskette ist es erforderlich, dass `cacert.pem` als vertrauenswürdiges Zertifikat bekannt ist – unter Windows kann es z.B. in den lokalen Keystore importiert werden – im Falle eines Datafox Geräts muss es in der Zertifikatsspeicher im Gerät übertragen werden.

Hinweis:
Selbstverständlich können Sie nun beliebig viele Zertifikate, die vom CA-Root-Zertifikat abgeleitet sind, erzeugen. Da OpenSSL diese intern verwaltet, ist es erforderlich, dass dieses ich hinsichtlich des Subjects unterscheiden.

B.7: Analyse von Zertifikaten

Microsoft Windows bietet mit dem `CertUtil` die Möglichkeit, den Inhalt der generierten Zertifikatsdateien zugänglich zu machen. Dazu ist es erforderlich, die Zertifikatsdatei zunächst zu dekodieren – anschließend kann deren Inhalt ausgegeben werden:

```
> CertUtil -decode my.cert my.crt
Eingabelänge = 1440
Ausgabelänge = 1021
CertUtil: -decode-Befehl wurde erfolgreich ausgeführt.

> CertUtil my.crt
X.509-Zertifikat:
Version: 3
Seriennummer: c4124040d28438f6
Signaturalgorithmus:
    Algorithmus Objekt-ID: 1.2.840.113549.1.1.11 sha256RSA
    Algorithmusparameter:
    05 00
Aussteller:
    E=s.meyer@datafox.de
    CN=Sven Meyer
    OU=Development
    O=Datafox
    L=Geisa
    S=Thueringen
    C=DE
  Namenshash (sha1): a12670bfda2ef055e608e130abab20741390a5d5
  Namenshash (md5): acc8642302e9f97b271419d7b06149eb

 Nicht vor: 27.09.2018 14:28
 Nicht nach: 27.09.2019 14:28

Antragsteller:
    E=s.meyer@datafox.de
    CN=Sven Meyer
    OU=Development
    O=Datafox
    L=Geisa
    S=Thueringen
    C=DE
  Namenshash (sha1): a12670bfda2ef055e608e130abab20741390a5d5
  Namenshash (md5): acc8642302e9f97b271419d7b06149eb

Öffentlicher Schlüssel-Algorithmus:
    Algorithmus Objekt-ID: 1.2.840.113549.1.1.1 RSA (RSA_SIGN)
    Algorithmusparameter:
    05 00
Länge des öffentlichen Schlüssels: 2048 Bits
Öffentlicher Schlüssel: Nicht verwendete Bits = 0
    0000  30 82 01 0a 02 82 01 01  00 ce 4b 2d 46 a8 05 75
    0010  73 d8 1c 88 49 97 64 0c  09 b0 96 0b 56 49 76 f0
    0020  1d 49 63 aa 80 cf 93 23  72 88 68 d6 ab 49 ba 7e
    0030  81 56 ae 57 21 d7 39 0b  f8 a1 e0 91 88 7e 9f d1
    0040  cb 32 ce c5 02 98 e0 e3  a2 17 0f c5 c1 0e 7a 57
    0050  d7 4b 11 16 b3 8a f5 ac  f1 b0 22 9f 75 4a e5 9a
    0060  9c 51 75 72 3b ea cb f3  94 6d 7e fb b0 d5 12 2d
    0070  1e e8 76 cf 70 42 69 94  71 89 34 f3 0c d7 bf 9a
    0080  ba 11 79 85 03 1d 46 01  00 2c 1a af ba 8c 7e 91
    0090  f2 a6 a0 d4 40 4e eb c6  10 6d 7a f9 3c f4 5f 1a
    00a0  55 77 20 19 6f 5c 42 76  44 51 ad a8 16 c1 3f e9
    00b0  96 0c 20 b7 f2 9f 6c 0e  7f 68 00 64 45 da 8b d3
    00c0  5c 2e 31 ed 63 01 cf 64  ea 52 d9 aa 44 b8 e9 15
    00d0  94 ea b0 2e 3a aa 5d 68  5a 13 d8 b1 de 68 2b f1
    00e0  7a a4 b8 ad 31 a8 f4 c3  62 20 ee 32 59 6e 33 6c
    00f0  1a 28 15 e9 13 27 e9 f6  18 94 44 cd 6b 64 b9 3d
    0100  a9 2c 9b c4 d0 1c 7b 77  71 02 03 01 00 01
Zertifikaterweiterungen: 3
```

```
    2.5.29.14: Kennzeichen = 0, Länge = 16
    Schlüsselkennung des Antragstellers
        480aacdb31e748625f02ae38aaab7a228722fb93

    2.5.29.35: Kennzeichen = 0, Länge = 18
    Stellenschlüsselkennung
        Schlüssel-ID=480aacdb31e748625f02ae38aaab7a228722fb93

    2.5.29.19: Kennzeichen = 0, Länge = 5
    Basiseinschränkungen
        Typ des Antragstellers=Zertifizierungsstelle
        Einschränkung der Pfadlänge=Keine

Signaturalgorithmus:
    Algorithmus Objekt-ID: 1.2.840.113549.1.1.11 sha256RSA
    Algorithmusparameter:
    05 00
Signatur: Nicht verwendete Bits=0
    0000  e0 63 94 47 a9 c0 e5 22  e2 ba 6e 7a 81 23 1f d7
    0010  91 96 94 77 0c 3d 40 33  e1 9f 4e 35 e6 f6 76 51
    0020  9e 45 1e b9 63 01 f4 6a  c4 04 06 5d a9 5c 10 be
    0030  b5 72 6e fd 0e ed 92 e7  eb 18 50 39 32 93 e2 55
    0040  1b 1d f4 a3 dd f3 28 6f  b0 fa 7f 88 85 9f 40 e0
    0050  90 9e 56 37 93 06 a6 0f  79 5e 9f f0 ef e4 36 55
    0060  85 a5 03 de aa 00 87 2b  b3 43 d1 20 14 51 ea a6
    0070  18 d8 a0 7d 8f 19 de 51  d5 54 02 c5 7a 92 39 52
    0080  84 ab 11 df b9 2f 78 9e  1f c5 f1 d9 b7 42 a6 0e
    0090  9a 84 3c 7f 56 05 81 c5  ac 4f 2e 99 39 77 88 84
    00a0  bd f2 c9 4b f0 a8 0b 58  83 bb d0 22 d4 5f 74 67
    00b0  45 5f 35 cf 90 0a 58 00  d1 05 60 38 ab 7b 0a 56
    00c0  2e 68 1c 4f 03 f6 7a 56  51 0a 38 65 a0 f2 e3 31
    00d0  c0 71 86 2e 06 d9 b0 a3  da 9a 23 45 5b 61 9e 1d
    00e0  7d 92 b0 1c b4 32 6d 80  e5 08 1e 14 05 9d 0d 40
    00f0  c7 c2 69 1b e8 81 d4 db  12 f5 36 77 e6 8e 27 80
Signatur stimmt mit dem öffentlichen Schlüssel überein.
Stammzertifikat: Antragsteller stimmt mit Aussteller überein
Schlüssel-ID-Hash(rfc-sha1): 480aacdb31e748625f02ae38aaab7a228722fb93
Schlüssel-ID-Hash(sha1): 09d0b7592a814746a0763cd728dadd7a63a6f3c7
Schlüssel-ID-Hash(bcrypt-sha1): 052b05f39aae0645b961e889c512889baa633aa0
Schlüssel-ID-Hash(bcrypt-sha256):
08beebf2f0d0b0cf4a857388dfb6a9ecda6f073665d7f44804e9552e16d469f1
Schlüssel-ID-Hash(md5): e3606281f405dc9bc9185609a419d76d
Schlüssel-ID-Hash(sha256):
29d0ee85e0290b1f3b13deda03a67bb824cc598c7e1ba4fbf5035f2290b8ecf1
Schlüssel-ID-Hash(pin-sha256): LSyEenteNDnDtS6o/57zWVbDOpCaOIOoyaNpcNfFuNQ=
Schlüssel-ID-Hash(pin-sha256-hex):
2d2c847a7b5e3439c3b52ea8ff9ef35956c33a909a3883a8c9a36970d7c5b8d4
Zertifikathash(md5): ce42b996362553fa26c594ad1ee81a24
Zertifikathash(sha1): fae481cd6b0e846ff5df4360844a013cac36d36c
Zertifikathash(sha256):
ea9dd651d5be4918bef5c699a444fca600129290ddc36def717b76004f0c2762
Signaturhash: 2f818c6fc579789dd464c4205eb8ceedf5568b483406fab4b9cae8b80e450684
CertUtil: -dump-Befehl wurde erfolgreich ausgeführt.
```

B.8: Grenzen der Implementierung

Die aktuelle Implementierung von HTTPS in Datafox Geräten weist Grenzen auf. Diese sind:

- Es werden nur TLS 1.1 und TLS 1.2 unterstützt.
- Die Implementierung bietet noch keine Unterstützung für TLS 1.3.
- Die Länge des RSA-Schlüssels darf nicht länger sein als 2048 Bit.

- Sollten dieses Ihren Sicherheitsanforderungen nicht genügen, setzen Sie bitte ECC mit einer Schlüssellänge von 256 Bit ein.

B.9: Weitere Informationsquellen

Zusätzliche Dokumentation zum Thema https können Sie z.B. im Netz an folgenden Stellen finden:

- https://tools.ietf.org/html/rfc2818
- https://de.wikipedia.org/wiki/Hypertext_Transfer_Protocol_Secure
- https://robertheaton.com/2014/03/27/how-does-https-actually-work/

Anhang C: Initiale Gerätekonfiguration über http

C.1: Versenden eines zyklischen Info-Telegrams mit Konfigurationsdaten

Mit den erweiterten Download-Möglichkeiten aus Anhang A bietet das http Protokoll alle notwendigen Funktionen zum Aufsetzen eines MasterIV Geräts.

Das Konzept erfordert eine Minimalkonfiguration des Geräts, die das Senden des Info-Telegrams (vgl. df_kvp=info in 2.2.3.2.12) ermöglicht. Das Info-Telegramm wird über Systemvariablen eingestellt und sorgt dafür, dass das Gerät sich nach der physikalischen Installation bei einem eingestellten Server meldet und von dort Konfigurationsdaten beziehen kann.

Das Verhalten wird dabei über die folgenden Systemvariablen geregelt:

Name der Systemvariablen	Beschreibung
http.config.mode	Zahl, die den Sende-Zyklus definiert 1 = täglich 2 = wöchentlich 4 = monatlich (28 Tage) Das erste Telegramm wird etwa 30 Minuten nach dem Systemstart gesendet.
http.config.host	Host, an den das Telegramm gesendet wird
http.config.port	Port des Hosts für den Telegramversand
http.config.send	URL auf dem Webserver für die Lieferung des Telegrams

Das Versenden des Telegramms erfolgt abhängig davon, ob auf dem Gerät beim Systemstart ein Setup vorliegt:

- Mit Setup: etwa 30 Minuten nach dem Systemstart, dann gemäß des in http.config.mode eingestellten Wiederholungsmuster
- Ohne Setup: etwa 30 Sekunden nach dem Systemstart („Notfallmodus“), dann etwa alle 10 Minuten.

Da das Gerät in diesem Modus bei konfiguriertem Aktive-Mode-Server auch den Aufbau einer Wartungsverbindung durchführt, kann es sein, dass Sie das Telegramm nicht exakt zur erwarteten Zeit erhalten – etwa wenn das Gerät gerade mit einem Wartungsserver verbunden ist oder die Verbindung dorthin aufzubauen versucht.

C.2: CRC Implementierung im Info-Telegramm

Das Info-Telegramm nutzt eine 32bit Prüfsumme, die mittels des CRC-Algorithmus berechnet wird. Dazu wird

- Initial: `0xffffffff`
- Polynom: `0xEDB88320`

eingesetzt, ein Reflect findet nicht statt.

Beispiel: Der CRC über die Zeichenkette „123456789“ beträgt `0x340BC6D9`.

C.3: Anwendungsfall: Monitoring und Aktualisieren von Zertifikaten im Gerät

Das zyklische Info-Telegramm kann z.B. eingesetzt werden, um die in Geräten hinterlegten Zertifikate „im Blick zu behalten“. Da das Info-Telegramm neben dem Gerättypen und der Seriennummer eine Liste von Zertifikatsdateien und deren Prüfsummen enthält, ermöglicht dieses, die eindeutige Erfassung der Zertifikate.

Folgende Fälle können auftreten:

- Sie planen keine Änderung an der Zertifikatskette des Servers und das Zertifikat, von dem Ihr Server-Zertifikat abgeleitet ist, ist noch „hinreichend lange“ gültig
    - Keine Aktion erforderlich
- Sie planen die Nutzung einer neuen Zertifikatskette auf dem Server
    - Übermitteln Sie das Zertifikat, von dem die neue Zertifikatskette abgeleitet ist, auf das Terminal.

Aus diesen Vorüberlegungen resultiert, wie Sie mit den Zertifikaten des Terminals verfahren wollen. Sie können diese entweder Löschen oder zusätzliche Zertifikate übermitteln. Das Szenario des Zertifikatsaustauschs wird eher nicht auftreten, da die Top-Level-Zertifikate von CA sehr lange gültig sind.

Das Löschen von Zertifikaten können Sie direkt per `df_remove_file` (siehe 2.2.3.2.17) veranlassen. Das gelöschte Zertifikat bleibt im Terminal noch so lange verfügbar, bis die Kommunikation neu initialisiert wird (etwa durch den Start des Wartungsmodus per `df_service` (siehe 2.2.3.2.1) oder den Neustart des Geräts).

Zum Übertagen eines Zertifikats verpacken Sie dieses in eine Transfer-Datei (Dateityp `0xDF00`, vgl. Anhang A.3.2), ordnen einen Namen zu, der mit „`.cert`“ endet und übermitteln die Transfer-Datei an das Terminal (sie können die IFF-Datei direkt als Antwort senden (vgl. 2.2.3.1.1) oder veranlassen das Herunterladen per `df_load_file` von Ihrem Webserver als nachgelagerte Aktion.

Anhang D: Test-Serveranwendung für die http-Integration

Zur Integration der http/https-Schnittstelle stellt Datafox einen Test-Webserver bereit. Mit diesem kann die Kommunikation zwischen einem Datafox-Gerät und dem Webserver ausprobiert werden und das Verhalten des Terminals auf Steuerkommandos beobachtet werden.

Achtung: Der Test-Server kommt ohne Support oder Garantie.

Hinweis:
Datafox stellt – Alternativ zum in diesem Kapitel vorgestellten Test-Server – eine Test-Umgebung im Internet zur Verfügung. Diese Test-Umgebung ist unter

https://www.datafox.de/support/testumgebungen

aus dem Internet zugreifbar.

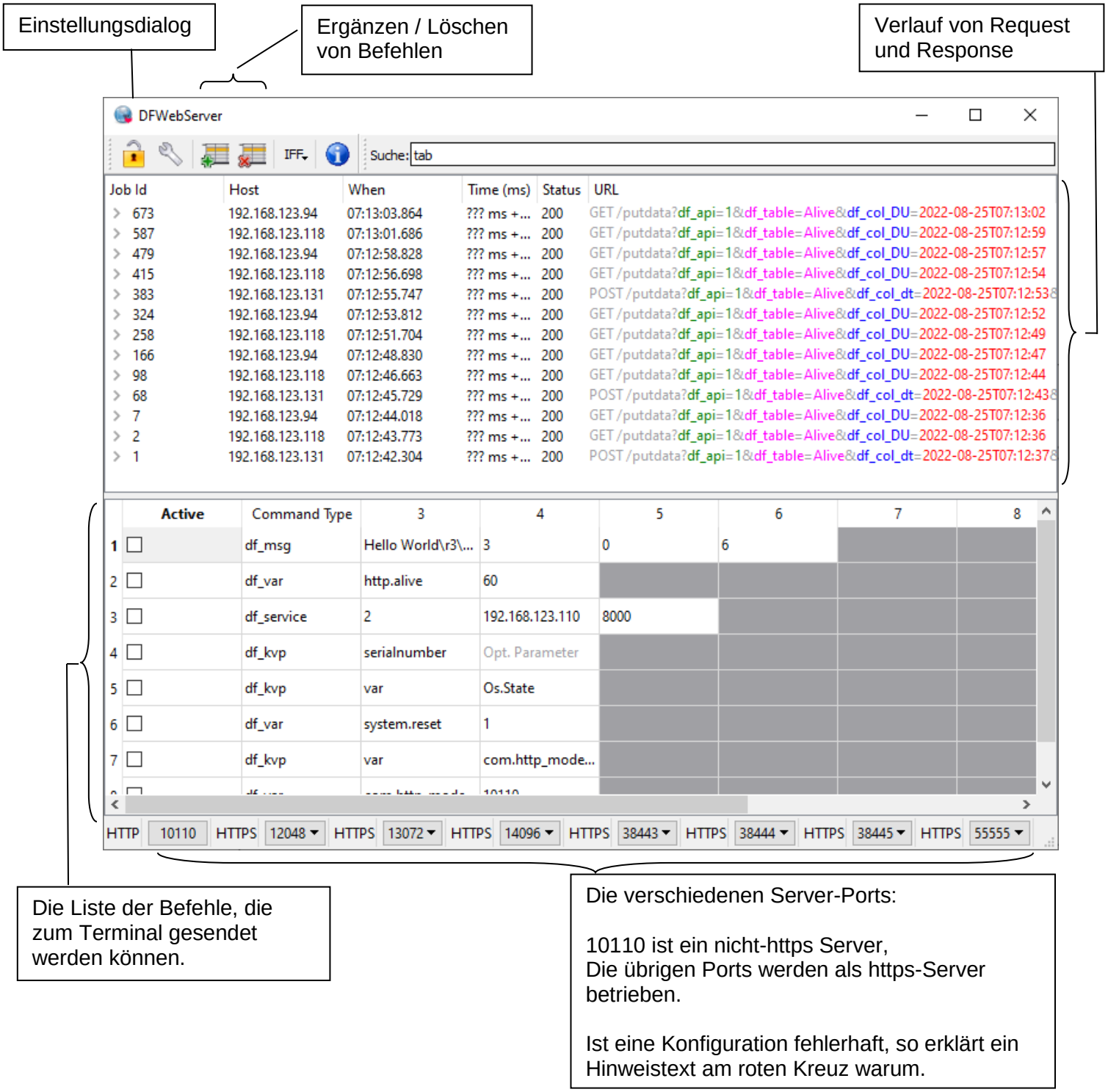
Der Testserver wird kontinuierlich erweitert. Die Referenz-Version steht unter

https://www.datafox.de/download/dist-DFWebServer-current.zip

zum Download bereit. Bitte beachte Sie, dass es sich bei der Anwendung um eine für interne Zwecke gedachte Test-Anwendung handelt, die ohne Gewährleistungsansprüche bereitgestellt wird.

D.1: Die Oberfläche

Nach dem Start stellt sich der Test-Server wie folgt dar:



Der Webserver nimmt auf allen korrekt konfigurierten Server-Ports Datensätze entgegen. Sie können die Konfiguration der Server-Ports über die Konfigurationsdateien settings-<*n*>.ini (n=0,…,9) steuern.

Die Settings-Dateien haben folgenden Aufbau:

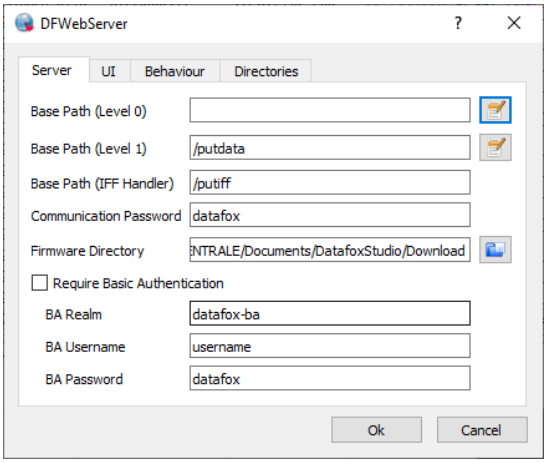
```
[General]
port=8443
minThreads=1
maxThreads=10
cleanupInterval=1000
readTimeout=60000
sslKeyFile=ssl/my.key
sslCertFile=ssl/my.cert
maxRequestSize=16000
maxMultiPartSize=1000000
```

Sofern Sie die Konfiguration des `sslKeyFile` und `sslCertFile` löschen, wird ein nicht-verschlüsselter http-Server erzeugt.

D.2: Konfiguration des Webservers

Im Einstellungsdialog erlaubt das Anpassen von Optionen des Servers, der Benutzeroberfläche, des Verhaltens und hinsichtlich der Nutzung von Verzeichnissen.

D.2.1: Server



Zur Konfiguration des Server-Verhaltens stehen folgende Parameter bereit:

Sie können unterschiedliche Anwendungspfade auf dem Webserver für API Level 0 (vgl. 2.1) und API Level 1 (vgl. 2.2) festlegen.

Der **Base-Path** stellt den Teil der URL auf dem Webserver dar, auf das Gerät zugreifen muss. Beginnt also im obigen Fall der Request nicht mit „/putdata“, so wird er nicht vom API Level 1 Server verarbeitet.

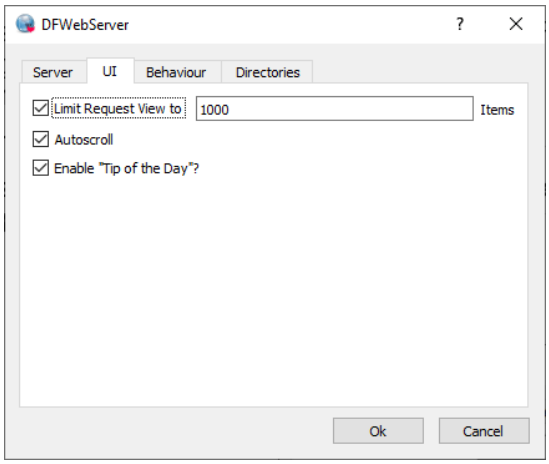
Der **Base-Path (IFF-Handler)** definiert den Pfad eines Endpunkts, an den IFF-Dateien vom Gerät gesendet werden können. Diese werden analysiert und ihre Struktur visualisiert.

Das **Kommunikationspasswort** wird für die RC4-basierte Verschlüsselung der Dateninhalte eingesetzt. Bitte nutzen Sie stattdessen TLS als Kanalverschlüsselung – RC4 ist nur aus Kompatibilitätsgründen zu sehr alten Implementierungen enthalten.

Firmware Directory verweist auf ein Verzeichnis, in dem DFZ-Dateien liegen. Das kann z.B. das Download-Verzeichnis des Datafox Studios sein.

Basic Authentication kann hier vorgegeben werden. Der Server verlangt dann als Zugangsvoraussetzung, dass das Gerät die hier eingestellten Anmeldedaten übermittelt.

D.2.2: Benutzerschnittstelle (UI)



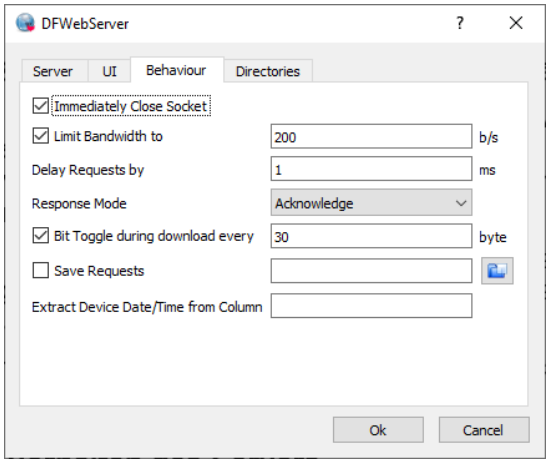
Im UI-Bereich können Sie das Verhalten der Oberfläche einstellen.

Sie können – aus Performance-Gründen – die **Anzahl der Requests**, die die Oberfläche darstellt, **begrenzen**. Wieviel hier sinnvoll ist, orientiert sich an Rechnerleistung und Ihrem Anwendungsfall. In vielen Entwicklungs-Situationen ist die Speicherung von Daten, die mehrere Stunden alt sind schlicht nicht notwendig.

Mit **Autoscroll** können Sie festlegen, ob ein selektierter Request in der Requestliste bei Eingang neuer Daten weiterhin dargestellt bleiben soll – oder aus dem sichtbaren Bereich gescrollt werden darf.

Der **Tip of the day** kann hier ab- bzw. angeschaltet werden.

D.2.3: Verhalten des Servers



Das Verhalten des Servers kann pro Request gesteuert werden:

Der Server kann die Aufforderung zum **Schließen der Verbindung** direkt mit seiner Antwort mitsenden. Dieses ist kein typisches Verhalten im http/1.1 Kommunikationsablauf – kann aber hier trotzdem zu Entwicklungszwecken eingestellt werden. Bitte beachten Sie, dass hier nicht

die Socket-Verbindung getrennt wird, sondern lediglich der „Connection: Close“-Header gesendet wird.

Wenn Sie die Funktion zum **Begrenzen der Bandbreite** aktivieren, sendet der Server seine Antwort-Pakete langsamer als die Hardware dieses zulassen würde.

Die **Verzögerung** zwischen dem Eingang der Anfrage und dem Aussenden der Antwort kann **in Millisekunden** eingestellt werden. Sie können auf diese Weise Systemlast auf dem Server simulieren.
Bitte beachten Sie, dass typischer Weise das TCP-Timeout die Verbindung nach etwa 20 Sekunden automatisch schließt und dann keine Antwort mehr von Server zum Gerät transportiert werden kann.

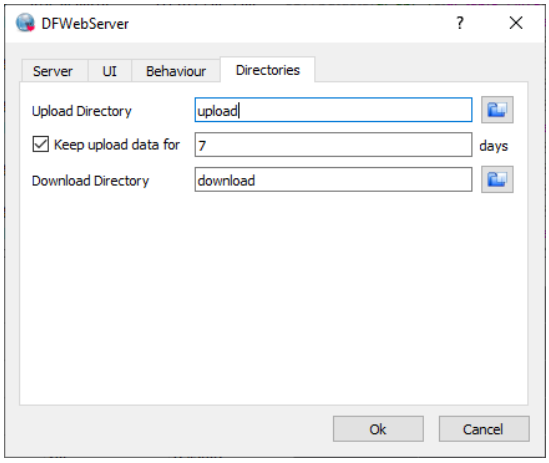
Ferner kann eingestellt werden, wie sich der Server generell verhalten soll. Typischer Weise sendet der Server eine **Bestätigung für den Nachrichteneingang**, hier kann ebenfalls ausgewählt werden, dass der Server überhaupt nicht antwortet (die Verbindung bleibt dann bis zum TCP-Timeout offen) oder bewusst kein Acknowledge sendet.

Zur **Simulation von Download-Fehlern** können Sie Fehler in Daten während des Downloads einbauen lassen. Diese Funktion sollten Sie nicht nutzen.

Um z.B. **Anfragen** für Ihre Anwendung **aufzeichnen** zu können, können Sie die Request-Daten des Geräts aufzeichnen lassen. Bitte beachte Sie, dass hier nur die Request-URL aufgezeichnet wird – der Body eines POST-Anfrage wird nicht erfasst.

Mittels der Option zur **Extraktion des Datums anhand einer Spalte** kann die Laufzeit einer Anfrage durch die Anwendung besser eingegrenzt werden. Der Server ermittelt anhand des hier eingestellten Spaltennamens den Erzeugungszeitpunkt des Datensatz (die Uhr des Geräts und des PCs müssen hinreichend synchron laufen)

D.2.4: Verzeichnisse



Mit den Verzeichnis-Einstellungen können Sie die Orte auf der Festplatte auswählen, die der Server für den Up-/Download-Zugriff benutzt. Auf http-Ebene werden die Zugriff auf diese Verzeichnisse per „/upload-area/“ bzw. „/download-area/“ bereitgestellt.

Zum **Hochladen von Daten** auf den Server wird der Upload-Pfad genutzt. Enthält dieser keine Laufwerks-Angabe, so wird er als relativer Pfad interpretiert. Der Inhalt des Verzeichnisses kann dann auf http-Ebene über „/upload-area/<dateiname>“ angesprochen werden.
Anmerkung: Bei der „/upload-area/“ handelt es sich um eine Definition der DFWebServer-Anwendung – in Ihrem Umfeld können Sie als Up-/Download-Pfade auf http-Seite natürlich andere URLs nutzen.

Der Webserver hat ferner eine Funktion, das Upload-Verzeichnis **regelmäßig aufzuräumen** und alte Dateien zu löschen. Wie alt eine Datei mindestens sein muss, können Sie vorgeben.

Analog zum Upload-Verzeichnis gibt es ein Verzeichnis im lokalen Dateisystem, aus dem das Gerät **Daten herunterladen** kann. Dieses wird auf http-Ebene als „/download-area/“ angesprochen.

Anmerkungen:

- Der Umgang des Test-Servers mit der Kommunikationsverschlüsselung per `df_cb` bzw. `df_ce` ist nur rudimentär implementiert.

D.3: Verarbeitung von Anfragen

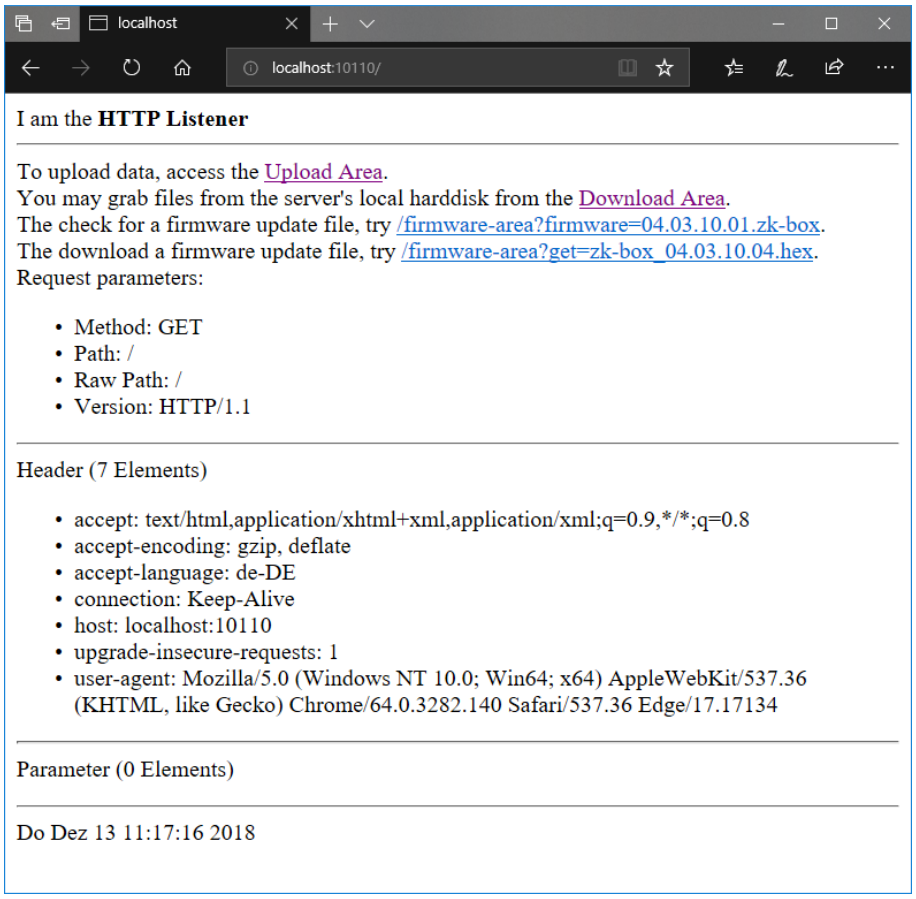
Die verschiedenen http- und https-Listener reagieren auf http-Anfragen. Sobald Sie ein Datafox Gerät mit den Verbindungsdaten des Servers ausgestattet haben und einen Datensatz erzeugen, werden Sie diesen im oberen Bereich des Servers erscheinen sehen.

Der Server wertet nun die Tabelle im unteren Bereich des Haupt-Fensters aus und setzt daraus die Antwort für das Gerät zusammen. Sie sehen im Screenshot oben, dass zusätzlich zur Quittung `df_api=1` in der Antwort ein `df_beep=1` (vgl 2.2.3.2) gesendet wurde.

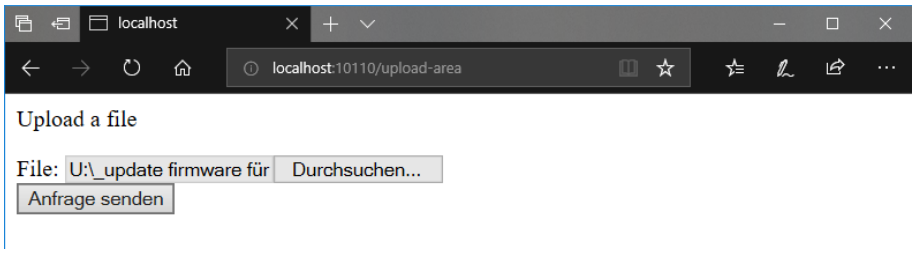
D.4: Werkzeuge für IFF-Dateien

D.4.1: Analyse von IFF Dateien

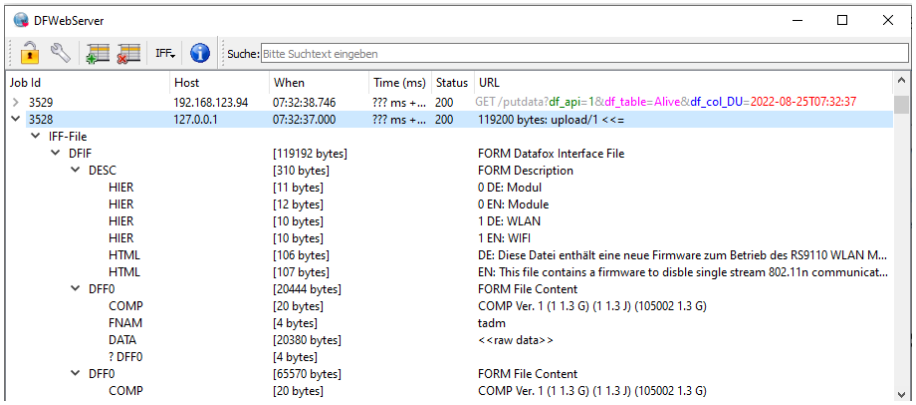
Der Web-Server ist für die Verarbeitung von IFF-Dateien vorbereitet. Diese werden typischer Weise vom Terminal auf `df_send_file`-Anforderungen (vgl. 2.2.3.2.16) hin übermittelt. Zum Experimentieren können Sie die Upload-Maske über einen Web-Browser bedienen. Greifen Sie dazu mit einem Web-Browser direkt auf einen den Listener-Ports zu:



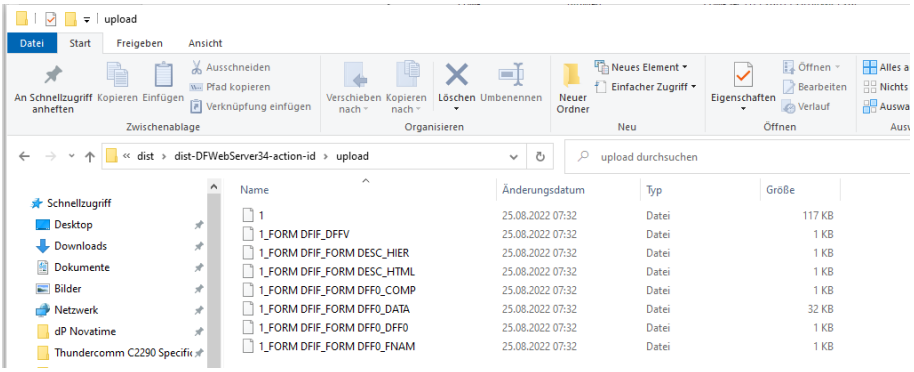
Die obige Seite ist eine interne Test-Seite des Web-Servers und kann für die Analyse von IFF-Dateien genutzt werden – egal ob diese vom Gerät oder von Ihnen für die Übertragung zum Gerät erzeugt wurde. Nutzen Sie dazu den Link „Upload Area“ und wählen eine IFF-Datei aus:



Nach dem Absenden der Anfrage wird diese Datei durch den Web-Server analysiert und wie folgt dargestellt:

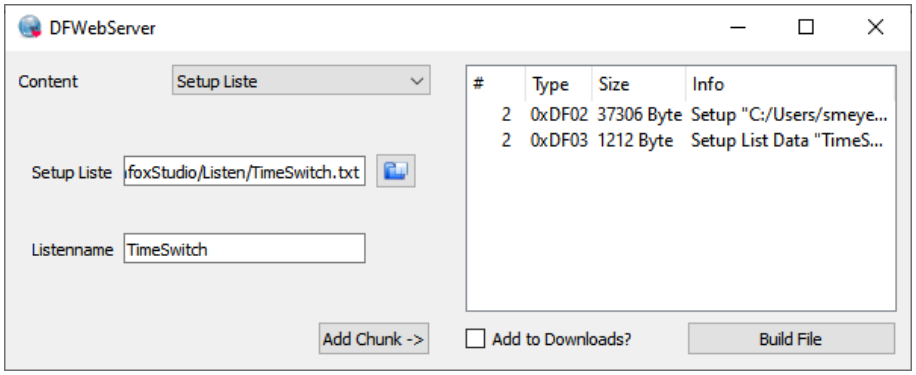


Den Inhalt der IFF-Datei trennt der Web-Server in seinem Upload-Verzeichnis für Sie wie folgt auf, wobei eingehende IFF-Dateien einfach von 1 starten durchgezählt werden:



D.4.2: Erzeugen von IFF Dateien

Der Webserver kann IFF-Daten für den Versand an Gerät zusammenstellen. Die Funktion ist über das „IFF“-Menü in der Toolbar verfügbar:



Der Prozess der IFF-Datei-Erzeugung ist zweigeteilt. Zunächst werden die Inhalte zusammengestellt und für die IFF-Erzeugung aufbereitet. Ist dieser Prozess abgeschlossen, kann die IFF-Datei erzeugt werden.

Auf der linken Seite wählen Sie den passenden Inhalt aus, der auf das Gerät übermittelt werden soll. Durch betätigen der Schaltfläche „Add Chunk ->“ wird dieser Inhalt dann für die Dateierzeugung übernommen und in die Liste auf der rechten Seite eingetragen.

Haben sie alle Inhalte zusammengestellt, können Sie per „Build File“ eine IFF-Datei erstellen lassen. Im Rahmen dieses Prozesses kann gleich eine Aktion in der Befehls-Tabelle des Webservers erstellt werden – aktivieren Sie dazu „Add to Downloads?“ vor dem Erzeugen der Datei.

D.5: Umgang und Aktualisierung der Server-Zertifikate

Der Server enthält im Auslieferungszustand eine Menge selbst-signierter Zertifikate, damit er möglichst einfach gestartet und genutzt werden kann. Hierbei handelt es sich sowohl im RSA- als auch ECC-Schlüssel/Zertifikats-Paare.

Sie können – falls die Zertifikate abgelaufen sind – mittels des Skripts „gen_certificates.sh“ einen neuen Satz Zertifikate erstellen – vorausgesetzt, Sie haben OpenSSL und eine Unix-Kommandozeilenumgebung (z.B. Cygwin) verfügbar.

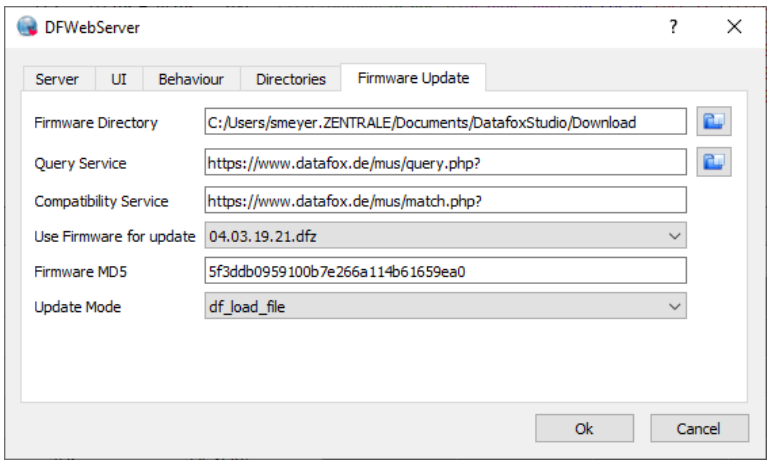
D.6: Firmware-Update über den Webserver

Der Test-Server implementiert auch einen Mechanismus zum Experimentieren mit dem Firmware-Update über HTTP. Dazu sind zwei Dinge erforderlich:

- Das Gerät muss ein „extinfo“-Telegramm an den Webserver senden
- Der Webserver muss für das Firmware-Update entsprechend vorkonfiguriert sein.

Konfiguration des Update-Service

Der Update-Service wird über einen eigenen Reiter im Konfigurationsdialog des DFWebServers eingerichtet. Eine Beispielkonfiguration sieht wie folgt aus:



- **Firmware-Directory** ist dasjenige Verzeichnis, in dem Sie verfügbare DFZ-Dateien abgelegt haben. Hier können Sie z.B. das Download-Verzeichnis des Datafox Studios nutzen.
- Der **Query-Service** ist Bestandteil des im Anhang E beschriebenen Mechanismus zum Auffinden einer kompatiblen Firmware-Version.

  Für eigene Entwicklungszwecke und zur Demonstration haben wir diesen auf unserem Webserver hinterlegt.

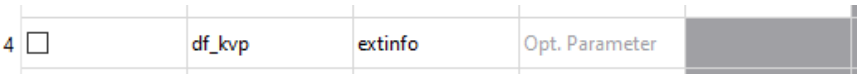
  Bitte nutzen Sie diesen Service **nicht** für Ihre Entwicklungszwecke oder gar den produktiven Einsatz.
- Der **Match-Service** ist ebenso wie der Query-Server Bestandteil des im Anhang E beschriebenen Mechanismus zum Auffinden einer kompatiblen Firmware-Version.

  Bitte nutzen Sie diesen Service **nicht** für Ihre Entwicklungszwecke oder gar den produktiven Einsatz.
- Sobald Query- und Match-Service eingerichtet sind, können Sie über den Button rechts vom Query-Service die Auswahl-Liste „**Use Firmware for update**“ mit Daten füllen. Anhand der ausgewählten Firmware-Version sucht der Server im „Firmware Directory“ nach der zugehörigen DFZ-Datei und ermittelt deren MD5-Hash („**Firmware MD5**“)
- Am Schluss der Konfigurationsparameter können Sie den **Update-Modus** auswählen. Hier stehen folgende Möglichkeiten zur Verfügung:

- „Direct IFF Data provisioning“:
  Die Firmware wird direkt im Response als IFF-Paket gesendet.
- „df_load_firmware“:
  Der Update-Service sendet denn „df_load_firmware“-Befehl. Dieser bezieht dann von einem explizit im Gerät hinterlegten Firmware-Update-Server die Daten der Firmware.
- „df_load_file“:
  Der Update-Service sendet einen „df_load_file“-Befehl. Dieser führt dazu, dass ein Download-Request an den Webserver selbst sendet. Über diesen stellt der Webserver dann die Daten der Firmware bereit.

Anfordern des „extinfo“ Telegrams

Sind die oben beschriebenen Einstellungen im Webserver getroffen, so kann per Eintrag in der Kommando-Liste die „extinfo“ des Geräts angefordert werden.

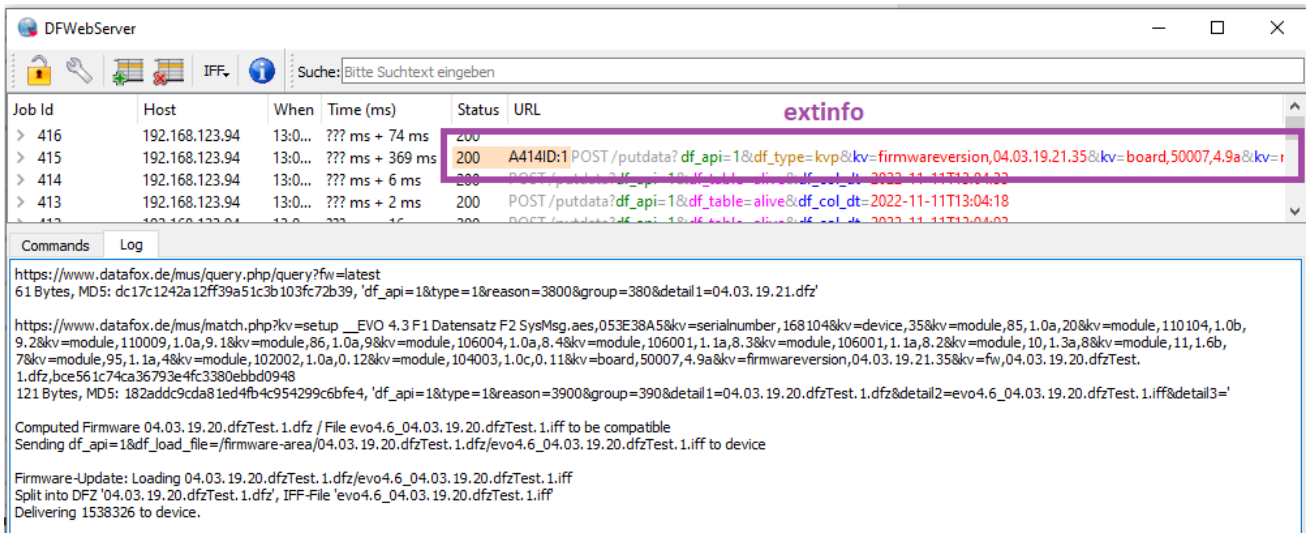


Die „extinfo“-Antwort des Geräts startet den Firmware-Update-Workflow:

- Prüfen der Kompatibilität von Gerät und gewünschter Firmware.
- Auslieferung der Firmware gemäß Update-Einstellungen.

Protocol

Die Teilschritte des Prozesses werden im Tab „Log“ aufgeführt:



Anhang E: Firmware-Update über HTTP(S)

Datafox Geräte implementieren ab Firmware Release 04.03.19.23 das Firmware-Update über HTTP(S). Dabei lädt das Gerät die „richtige“ Firmware-Datei herunter, prüft dieses und – falls die Firmware das Gerät unterstützt – installiert diese.

Das Auffinden der „richtigen“ Firmware ist Gegenstand dieses Anhangs.

Zur Orientierung:

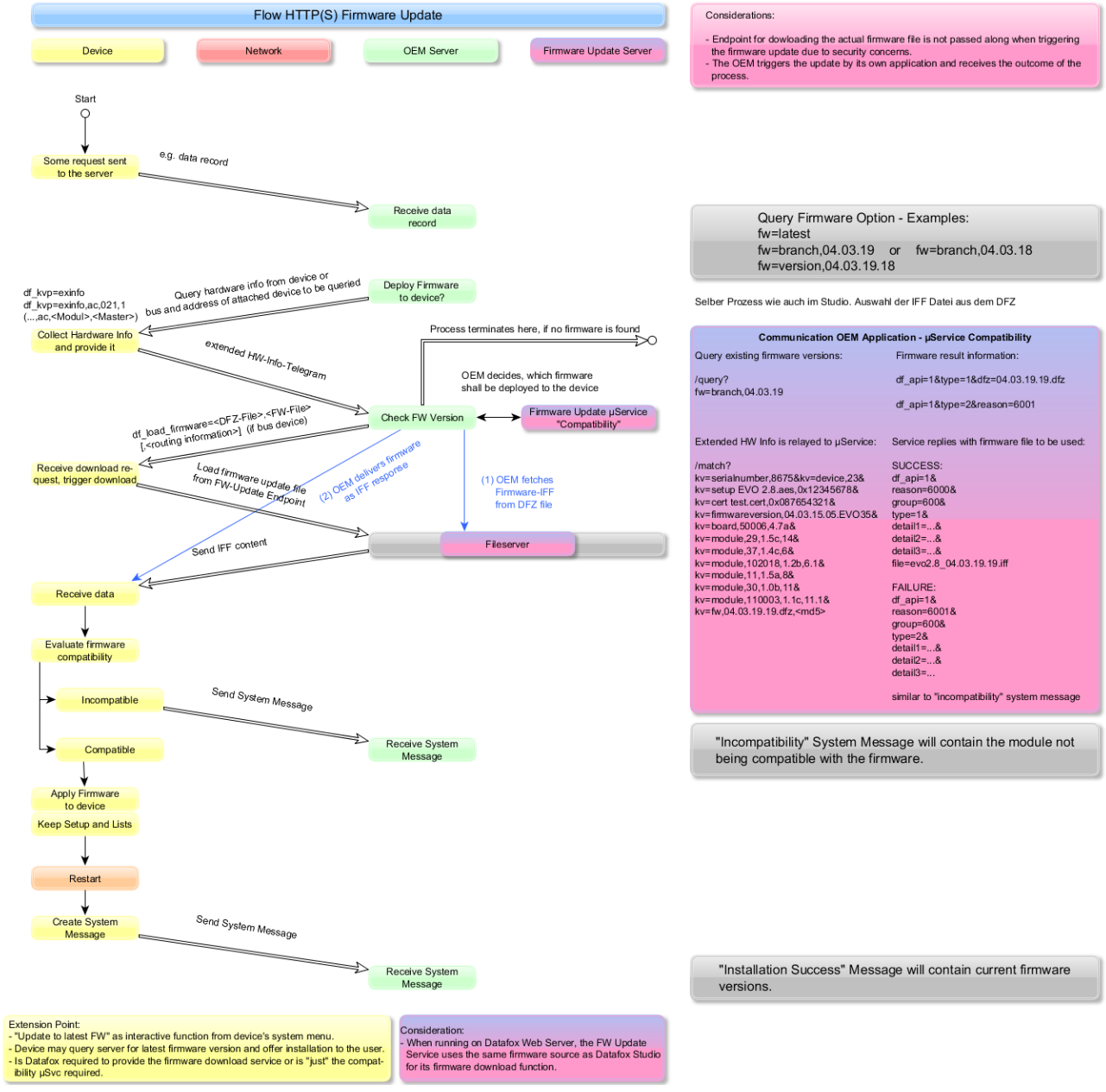


Diagramm zum Ablauf des Firmware-Updates

Der Ablauf des Firmware-Updates erfolgt in Zusammenarbeit zwischen dem Datafox Gerät, der OEM-Anwendung, den Skripten „query.php und/oder „match.php“ sowie einer Datenquelle für Firmware-Dateien.

Dabei übermittelt das Gerät zyklisch Datensätze oder Alive-Meldungen. Diese können genutzt werden, um vom Gerät eine „Selbstauskunft“ (extended info, „extinfo“) anzufordern. Die „extinfo“ enthält Informationen über im Gerät eingesetzte Hardware-Komponenten, die für die Kompatibilitätsprüfung erforderlich sind.

"match.php“ kann prüfen, ob eine Firmware-Version zum Satz der übermittelten Geräte-Hardware passt und berechnet, welche IFF-Datei die korrekte Firmware des Geräts enthält.

Zur Übermittlung der Firmware stehen nun drei Wege zur Verfügung:

- Die OEM-Anwendung kann die Firmware direkt im Response senden (vgl. Abschnitt 2.2.1.2)
- Die OEM-Anwendung teilt dem Gerät mit, dass dieses eine IFF-Datei (von einem dedizierten Firmware-Update-Server) herunterladen soll (df_load_firmware, vgl. Abschnitt 2.2.3.2.25).
- Die OEM-Anwendung teilt dem Gerät mit, dass es eine IFF-Datei mit Firmware-Inhalt herunterladen soll (df_load_file, siehe 2.2.3.2.15)

Die beiden Skripte erwarten die Anfragedaten ins klassischer URL-Kodierung mit der GET-Anfrage, als Antwort wird ein URL-kodierter Datensatz, der einer Systemmeldung nachempfunden ist, geliefert.

Beispiel:

Anfrage:

```
…/query.php?fw=latest
```

Antwort:

```
df_api=1&type=1&reason=3800&group=380&detail1=04.03.19.20.dfz
```

Hinweis:
Die Struktur des Antwort-Datensatz ist in den PHP-Skripten eingebaut. Diese wird nicht durch ein etwaiges Setup beeinflusst.
Der generelle Aufbau von Systemmeldungen ist unter 2.3 beschrieben.

E.1: Voraussetzungen für den Einsatz von „query.php“ und/oder „match.php“

Sowohl „query.php“ als auch „match.php“ erwarten, dass die verfügbare Geräte-Firmware im lokalen Dateisystem des Servers vorhanden und zugänglich ist. Um effizient den MD5-Fingerprint der DFZ-Datei prüfen zu können, muss dieser ebenfalls vorberechnet sein.

Die Firmware muss dabei unter dem Namen des DFZ entpackt werden, zusätzlich wird eine Datei „<Name des DFZ>.md5“ erwartet, in der die Hexadezimal-Darstellung des MD5-Hashes der DFZ-Datei enthalten ist.

Bezogen auf ein Wurzelverzeichnis $ROOT gehören als zur Firmware-Version 04.03.19.19 mindestens folgende Dateien:

```
$ROOT/04.03.19.19.dfz/*.iff
$ROOT/04.03.19.19.dfz.md5
```

Andere Dateien des DFZ-Archivs, wie etwa die „*.hex“-Dateien, werden im Rahmen des Firmware-Updates nicht genutzt.

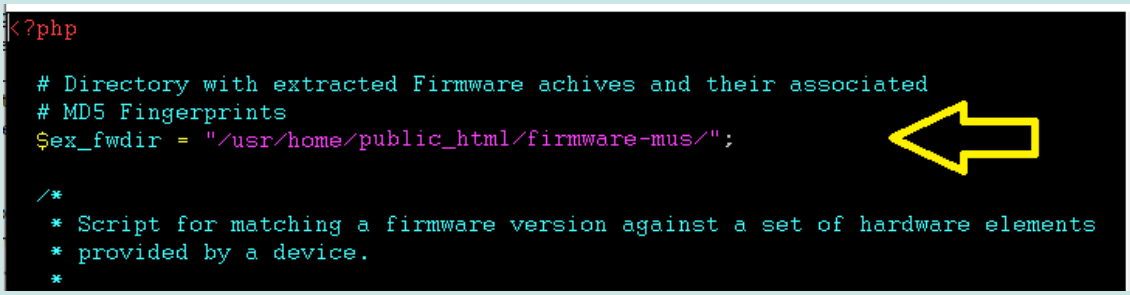
E.1.1: Beispiel-Skript zum Hinterlegen einer Firmware-Version auf dem Server

Zum Auspacken einer Firmware-Datei kann z.B. folgendes Skript (unter Linux) genutzt werden. Bitte beachten Sie, dass das Skript hier ein Beispiel-Skript ist, das „AS-IS“ bereitgestellt wird und ohne jegliche Garantie bereitgestellt wird:

```
#!/bin/bash

input_fn=$1
base_fn=`basename $input_fn`

rm -rf $base_fn $base_fn.md5
mkdir $base_fn
( cd $base_fn && unzip ../$input_fn > /dev/null )

md5sum $input_fn | cut -b 1-32 > $base_fn.md5

echo IFN $input_fn
echo BFN $base_fn
```

Es wird aus dem $ROOT-Verzeichnis mit dem Pfad zu einer DFZ-Datei aufgerufen, entpackt die DFZ-Datei in ein entsprechendes Unterverzeichnis und legt die md5-Datei an.

Hinweis:
Der Pfad zum $ROOT-Verzeichnis ist Bestandteil der beiden Skripte. Bitte tragen Sie den Ort, an dem die Firmware-Daten auf Ihrem Server liegen, beim Deployment in beide Skripte entsprechend ein:



Bezugsquelle für die PHP Skripte:

https://datafox.de/download/musvc_firmware_update.zip

E.2: Funktionsweise „query.php“

Das Skript „query.php“ liefert Informationen zu auf dem Server vorhandenen Firmware-Versionen. Die Parameter des Skripts werden als Anfrage-Parameter erwartet, als Resultat gibt das Skript eine Systemmeldung zurück.

E.2.1: Ermitteln der neusten Firmware-Version

Ermitteln der neusten, auf dem Server hinterlegten Firmware-Version.

Anfrage:

…/query.php?**fw=latest**

Antwort:

```
df_api=1&type=1&reason=3800&group=380&detail1=04.03.19.20.dfz
```

E.2.2: Ermitteln der neusten Firmware-Version eines Release-Zweiges

Ermitteln der neusten Firmware-Version aus einem Release-Zweig.

Anfrage:

…/query.php?**fw=branch,04.03.19**

Antwort:

```
df_api=1&type=1&reason=3800&group=380&detail1=04.03.19.20.dfz
```

Anfrage:

…/query.php?**fw=branch,04.03.16**

Antwort:

```
df_api=1&type=1&reason=3800&group=380&detail1=04.03.16.06.dfz
```

E.2.3: Prüfen, ob eine bestimmt Firmware-Version auf dem Server vorhanden ist

Ermitteln, ob eine bestimmte Firmware-Version auf dem Server vorhanden ist.

Anfrage:

…/query.php?**fw=version,04.03.19.20**

Antwort:

```
df_api=1&type=1&reason=3800&group=380&detail1=04.03.19.20.dfz
```

Anfrage:

…/query.php?**fw=version,04.03.19.21**

Antwort:

```
df_api=1&type=2&reason=3802&group=380&detail1=no match
```

E.2.4: Auflisten aller vorhandenen Firmware-Versionen

Ermitteln einer Liste aller, auf dem Server vorhandenen Firmware-Versionen

Anfrage:

…/query.php?**fw=list**

Antwort:

```
df_api=1&dfz=04.03.19.20.dfz&dfz=04.03.19.19.dfz&…&dfz=04.03.19.01.dfz&dfz=
04.03.18.08.dfz&dfz=04.03.16.06.dfz&…&dfz=04.03.09.20.dfz&dfz=04.02.05.60.d
fz
```

E.3: Funktionsweise „match.php“

Das Skript „match.php“ berechnet, ob eine gewünschte Firmware-Version mit einem Gerät kompatibel ist und liefert dann diejenige Datei des Firmware-Archivs, die an das Gerät zum Update geliefert werden muss.

Die wesentlichen Informationen für die Anfrage liefert dabei das Gerät selbst im Rahmen der „extinfo“-Selbstauskunft. Diese liefert (vgl. 2.2.3.2.12) Informationen über das Gerät, seinen Betriebszustand, die Hauptplatine und die eingebauten Module. Für die Anfrage an das „match.php“-Skript ist es erforderlich, dass Sie noch die gewünschte Firmware-Version und deren MD5-Prüfsumme ergänzen (in Gelb hinterlegt im folgenden Request).

Die Prüfsumme berechnen Sie bitte über die Ihnen vorliegende Version der DFZ-Datei – der „match.php“ prüft dann, ob ihm die gleiche Datei vorliegt.

```
.../match.php?
      kv=firmwareversion,04.03.20.03.Evo43&
      kv=board,50007,4.4a&
      kv=module,102026,1.0a,0.11&
      kv=module,104003,1.0a,0.12&
      kv=module,12,1.2a,1&
      kv=module,35,1.3a,2&
      kv=module,1,1.3j,6&
      kv=module,11,1.6b,7&
      kv=module,10,1.1c,8&
      kv=module,106002,1.0a,8.1&
      kv=module,106001,1.0a,8.2&
      kv=module,106001,1.0a,8.3&
      kv=module,106004,1.0a,8.4&
      kv=module,19,1.3a,9&
      kv=module,110004,1.0a,9.1&
      kv=module,110101,1.0c,9.2&
      kv=module,20,1.3a,18&
      kv=device,11&
      kv=serialnumber,1234&
      kv=setup EVO 4.3 F1 Datensatz F2 SysMsg.aes,0AF95295&
      kv=fw,04.03.20.03.dfz,8d3343de9a00cb36e7617e66ace126d8
```

Als Antwort erhalten Sie entweder eine Fehlermeldung (erkennbar an …&type=2&…) oder den Namen der auszuliefernden IFF-Datei aus dem Container.

Fehlermeldung bei der Kompatibilitätsprüfung:

```
df_api=1&type=2&reason=3911&group=390&detail1=no acceptable compatibility
info
```

Informationen zu einer passenden Firmware:

```
df_api=1&type=1&reason=3900&group=390&detail1=04.03.19.20.dfz&
detail2=evo_intera_II_04.03.19.20.iff
```

E.4: Auslieferung der Firmware-Dateien

Die Bereitstellung der Firmware-Inhalte erfolgt über Ihren OEM-Service, der die Kommunikationsverbindung zum Gerät hält. Über den blauen Pfeil im obigen Diagramm können Sie eine Netzwerk-Interaktion abkürzen, es ist dann allerdings erforderlich, dass die Bereitstellung der Daten direkt innerhalb der vom Gerät gestarteten HTTP(S) Session erfolgt.

Falls Sie einen Standard-Webserver nutzen möchten, dann können Sie das Gerät per `df_load_firmware` dazu anweisen einen File-Download-Request zum im Gerät hinterlegten Firmware-Update-Server zu senden. Dieser kann dann ein „normaler“ Webserver, der für die Auslieferung von Webseiten optimiert ist.

Anhang T: Troubleshooting

Dieses Kapitel fasst Aspekte zusammen, die Datafox im Rahmen von Inbetriebnahme-Szenarien beobachtet hat. Selbst wenn die hier beschriebenen Szenarien nicht genau zu einem aktuellen Problem passen, das Sie beobachten, können diese dennoch als Anregungen für die Forschung nach Fehlerursachen genutzt werden.

T.1: Probleme mit spezifischen Webservern

Wir haben beobachtet, dass nicht alle Web-Server in RFC-Verträglicher Weise implementiert sind. Bislang sind wir in diesem Umfeld auf folgende Probleme gestoßen, die wir über die Systemvariable `http.flags` lösbar machen.

T.1.1: Port im Host-Header des http-Requests

Die RFC 2616 (http/1.1) definiert den Host-Header eines http-Requests als

```
Host = "Host" ":" host [ ":" port ]
```

Es gibt Web-Server, die Requests mit Port im Host-Header nicht verarbeiten und die Requests zurückweisen (oder sogar abstürzt). Andererseits gibt es Reverse-Proxies, die den Host-Header mit Port benötigen, um den korrekten Rechner auszuwählen, an den der Request weitergeleitet werden soll.

Falls Sie einen Web-Server haben, der keinen Port im Host-Header verarbeiten kann, setzen Sie bitte Bit 0 in den `http.flags`.

!

Achtung:
Ein Server, der HTTP Requests mit Port im Host-Header nicht verarbeitet, kann an unterschiedlichen Stellen bei der Informationsverarbeitung vorkommen, insbesondere bei Cloud-Lösungen.

Wir haben eine Cloud-Lösung analysiert, die – nach dem Aufbau der verschlüsselten Verbindung – auf Ebene des Applikationsprotokolls einen Request per http 502 Bad Gateway beendete. Da die SSL-Kommunikation zuvor korrekt hergestellt wurde, muss die Antwort von einem vermittelnden Server gesendet worden sein, der den Ziel-Host aufgrund des Ports im HOST-Header nicht zuordnen konnte.

T.1.2: Proxy-Server/Load-Balancer und Wartungsmodus (Connection: Close)

Der Aufbau der Wartungsverbindung durch ein Gerät (vgl. `service` (2.1.2.2.1) bzw. `df_service` (2.2.3.2.1)) erfordert das Schließen der bestehenden http-Verbindung entweder durch Timeout oder Aufforderung des Servers. Die Aufforderung des Servers wird per http-Header „`Connection: close`“ übermittelt.

Falls Sie einen Proxy-Server einsetzen, so interpretiert dieser das „`Connection: close`“ im http-Header und schließt die Verbindung zwischen Ihrem Server und Proxy, nicht aber zwischen dem Proxy und dem Gerät. Das Gerät baut entsprechend keine Wartungsverbindung auf.

Sie können das Gerät bei Eingang eines Service-Requests per http zum expliziten, eigenen Schließen der Verbindung veranlassen, indem Sie das Bit 1 in den `http.flags` setzen.

T.1.3: Request mit absoluteURI bzw. abs_path

Die RFC 2616 (http/1.1) definiert in Abschnitt 5.1.2 den Aufbau einer Request-URI in der ersten Zeile des http-Requests als

```
Request-URI    = "*" | absoluteURI | abs_path | authority
```

Die erste Zeile eines `absoluteURI`-Requests sieht wie folgt aus:

```
GET http://www.w3.org/pub/WWW/TheProject.html HTTP/1.1
```

Ein `abs_path`-Request wird ohne Protokoll und Host in der Request-Zeile erzeugt. Der Host wird dann über den begleitenden Host-Header definiert:

```
GET /pub/WWW/TheProject.html HTTP/1.1
Host: www.w3.org
```

Für http-Requests ist die `abs_path` Form normal, `absoluteURI` wird aufgrund der Kompatibilität zu Proxy-Servern für einen http/1.1-Server ebenfalls verlangt. Da es Reverse-Proxies gibt, die anhand des `absoluteURI` den Ziel-Host auflösen, haben wir uns entschlossen, diese Form in der Firmware zu implementieren.

Leider versteht nicht jeder Web-Server diesen Request-Typ. Falls Sie einen Web-Server einsetzen, der keine `absoluteURI`-Requests verarbeitet, setzen Sie bitte in den `http.flags` das Bit 2.

T.1.4: Beispiel: Setzen der richtigen http.flags

Wie zuvor bemerkt, handelt es sich bei den `http.flags` um ein Bitfeld, das als Zahlenwert (Dezimalzahl) an das Gerät übergeben wird. Sollte Ihre Installation folglich die `abs_path`-Adressierung benötigen und zusätzlich Probleme mit dem Port im Host-Header haben, ist es erforderlich, die Bits 0 und 2 in den `http.flags` zu setzen.

Dazu müssen Sie die `http.flags` auf den Wert $5 = 2^0 + 2^2$ einstellen.

T.2: HTTPS-Verbindungsaufbau zu AWS / CloudFront

Amazon CloudFront benötigt einen korrekt gesetzten SNI-Hostname, sonst bricht der HTTPS-Handshake mit einem Alert 40 ab. Der SNI-Hostname ist identisch mit dem Hostnamen, der Ihrer AWS-Instanz zugeordnet ist.

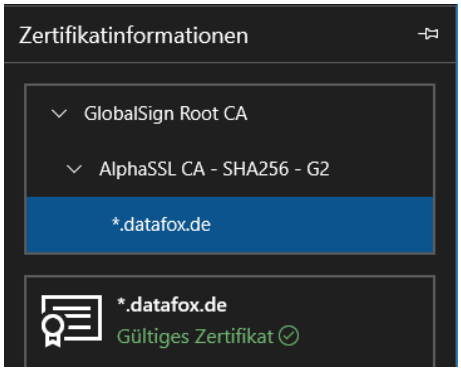
Bitte tragen Sie diesen für die Gerätekommunikation als `com.http_mode[.].tls.sni_host` ein.

T.3: Mein Gerät meldet SSL-Write -9984 Fehler – obwohl das Zertifikat korrekt auf dem Gerät hinterlegt ist

Gemäß TLS-Spezifikation soll das Gerät die vollständige Zertifikatskette exklusive des Root-Zertifikats vom Server erhalten. Zusammen mit dem Root-Zertifikat ist diese Kette dann prüfbar.

Wir haben allerdings Server beobachtet, die nicht die vollständige Zertifikatskette übermitteln. In diesem Fall müssen dann **auch** die Intermediate-Zertifikate als vertrauenswürdige Zertifikate auf dem Gerät hinterlegt werden.

Betrachtet man das Beispiel der „datafox.de“ Webseite, so steht in der Zertifikatsansicht zwischen dem „*.datafox.de“ und dem „GlobalSign Root CA“ noch das „AlphaSSL CA – SHA256 – C2“ Zertifikat.



Exportieren und konvertieren Sie dieses bitte wie in Anhang B beschrieben und übermitteln es ebenfalls an das Gerät.

T.4: Virtual Hosts und HTTPS (z.B. Microsoft IIS)

Im HTTP Protokoll erfolgt das Auflösen von virtuellen Hosts (als mehreren logischen Servern, die auf einem physischen Server betrieben werden) in der Regel über den HOST-Header des HTTP Requests.

Beim Verbindungsaufbau per HTTPS stehen die Header des HTTPS-Requests während des SSL Handshakes nicht bereit – als weiteres Problem kommt hinzu, dass sich das Zertifikat je Server-Instanz unterscheiden wird.

Sollten Sie eine Installation mit virtuellen Hosts nutzen, setzen Sie bitte den SNI-Hostnamen. Nach unseren Erkenntnissen nutzt zumindest der Microsoft Internet Information Server diesen, um das korrekte Zertifikat und den korrekten, virtuellen Host zuzuordnen.

T.5: Referenz http.flags

Über die http Flags können folgende Einstellungen vorgenommen werden:

Bit / Wert	Name	Beschreibung
1 / 1	Host-Header ohne Port	Ist dieses Flag gesetzt, so wird der Host-Header ohne Port gesendet (vgl. E.1.1)
2 / 2	Close bei Service	Ist dieses Flag gesetzt, so schließt das Gerät eine bestehende http-Verbindung aktiv, sobald ein Service-Request vom Web-Server übermittelt wurde (vgl. E.1.2)
~~3 / 4~~	~~Nutzung von abs_path-Request~~	~~Ist dieses Flag gesetzt, so werden die Requests nicht mehr mit absoluteURI sondern mit abs_path generiert (vgl. E.1.3)~~

		Achtung: Dieses Flag ist mit Version 04.03.18.03 obsolet. Die Entscheidung abs_path / absoluteURI wird jetzt bezogen auf den Proxy-Server getroffen: - Ist ein Proxy-Server eingetragen, so wird absoluteURI genutzt. - Ist kein Proxy-Server eingetragen, so wird abs_path eingesetzt.
4 / 8	Ohne Static-Header	Ist dieses Flag gesetzt, so werden lediglich der `Host:`- und der `Content-Length:`-Header generiert. Alle übrigen Header entstammen dann den „header.extensions“ (vgl. 1.7) oder bleiben leer.
5 / 16	absoluteURI erzwingen	(ab 04.03.18.03): Ist dieses Flag gesetzt, so wird die absoluteURI-Adressierung unabhängig davon genutzt, ob ein Proxy-Server eingetragen ist.

T.6: Basic Authentication funktioniert nicht

Wir haben beobachtet, dass Basic Authentication in Verbindung mit M2M-Mobilfunk-Karten nicht immer funktioniert. Nach unseren Untersuchungen liegt dieses daran, dass der Mobilfunk-Anbieter die TCP/IP-Verbindung direkt nach dem Zustellen der Daten des Pakets trennt.

In diesem Szenario ist es nicht möglich, den BA-Realm zu ermitteln: Das Gerät speichert den Realm nur für die Dauer einer bestehenden Verbindung aus Sicherheitsgründen. Da mit dem Erhalt der http-401 Unauthorized-Antwort zur Realm-Ermittlung die Verbindung getrennt wird, wird konsequent die Realm-Information wieder verworfen und der nächste Verbindungsaufbau wird ohne Basic Authentication Informationen durchgeführt (das Gerät kennt den Realm nicht).

T.7: Unterschiedliche Zertifikate/Zertifikatsketten auf demselben Webserver

Aktuell unterstützen wir mit der Geräte-Firmware RSA-Zertifikate mit bis zu 2048 Bit Schlüssellänge. Diese Einschränkung gilt für alle Zertifikate in der Zertifikatskette.

Die Nutzbarkeit von Zertifikatsketten mit längeren Schlüsseln hängt maßgeblich an der Kommunikationsart und Geräteausstattung des Geräts – daher ist diese nicht generell garantiert.

Eine Möglichkeit, auf einem Webserver gezielt unterschiedliche – und damit auch unterschiedlich lange – Zertifikat einsetzen zu können, bieten Virtuelle Hosts. Sie können die Nutzung eines speziellen virtuellen Hosts über den SNI_HOST festzulegen, falls dieser nicht im DNS vorhanden ist, und so einen dedizierten Endpoint auf Ihrem Webserver erstellen, der eine Zertifikatskette mit 2048 Bit einsetzt.

T.8: Freier Speicher bei aktiver TLS-Kommunikation

Datafox Geräte protokollieren freien Speicher im Rahmen der „12 Stunden-Statistik“, die um Mitternacht und mittags – bezogen auf die Uhrzeit des Geräts – erstellt und im Systemlog dokumentiert wird.

2023-03-20 12:00:00	742218 | FW <EVO46.04.03.20.08, 2023-03-02, 16:06:56>, OPERATING TIME <TOTAL 558d:17h (2021-08-18|96), BOOT 05h:15m:05s (2023-03-20,06:44)>.
2023-03-20 12:00:00	742219 | HEAP <ALLOCS 1633355 (120.470.996), OBJECTS 55026 (65535), TOTAL 141544, USED 106904, FREE 34640, MIN 26772, NULL 0>.
2023-03-20 12:00:00	742220 | FLASH <SP/f (48%|4), E1897, F239, L3265, Ex297.153.889, W(u147|p23701|m6378), R0|2309, W0|80>.
2023-03-20 12:00:00	742221 | RECORDS <TOTAL 4194304, COUNT 0 (0,0%), WRITE 1 (154), READ 2 (308), W0, P0, C0, H0>.

Die Zeile „HEAP“ dokumentiert, dass

- Insgesamt 141544 Bytes Speicherplatz zur Verfügung stehen (TOTAL)
- Davon sind aktuell 106904 Byte belegt (USED)
- Entsprechend sind 34640 Byte aktuell frei (FREE)
- Während des Betriebs wurde der Speicher bis auf 26772 Byte genutzt (MIN) und
- Die Speicherbelegung schlug bislang 0 mal fehl (NULL).

Bitte achten Sie darauf, dass das Gerät bei aktiver HTTPS-Kommunikation mindestens 16 kB im Bereich MIN aufweist.

Hinweis:
Bitte achten Sie darauf, dass die Speicherbelegung bei laufender HTTPS-Kommunikation erfolgt. Sobald Sie z.B. ein USB-Kabel am Gerät anschließen, wechselt dieses die Kommunikationsart, die Speicherbelegungswerte FREE und USED passen dann nicht um HTTPS-Szenario.