

T&D 製データロガー TR-7wf/nw プロバイダ

Version 1.0.0

ユーザーズ ガイド

June 26, 2019

備考:

【改版履歴】

バージョン	日付	内容
1.0.0	2019-7-8	初版

【動作確認機器】

機種	注意事項
TR-72nw(日本語版のみ)	

目次

1. はじめに.....	5
2. プロバイダの概要	6
2.1. 概要	6
2.2. メソッド・プロパティ	8
2.2.1. CaoWorkspace::AddController メソッド.....	8
2.2.2. CaoController::GetVariableNames プロパティ.....	9
2.2.3. CaoController::AddVariable メソッド.....	9
2.2.4. CaoController::Execute メソッド	9
2.2.5. CaoController::AddExtention メソッド.....	10
2.2.6. CaoController::OnMessage イベント	11
2.2.7. CaoExtention::AddVariable メソッド	12
2.2.8. CaoVariable::get_Value プロパティ.....	12
2.2.9. CaoVariable::put_Value プロパティ.....	12
2.3. 変数一覧.....	13
2.3.1. CaoController クラス.....	13
2.3.2. CaoExtention クラス	14
2.3.2.1. 動作設定値送信用.....	14
2.3.2.2. 動作設定値受信用.....	17
2.3.2.3. 現在値データ	21
2.3.2.4. 警報データ.....	24
3. コマンドリファレンス	27
3.1. CaoController クラス.....	27
3.1.1. CaoController::Execute ("Start") コマンド	27
3.1.2. CaoController::Execute ("Stop") コマンド	28
3.1.3. CaoController::Execute ("Restart") コマンド	28
4. サンプルプログラム	29
5. 補足.....	31
5.1. ACL 設定	31
5.1.1. 追加	31
5.1.2. 削除	31

5.1.3. 確認	32
5.2. Firewall 設定	33
5.3. スケール変換式.....	37

1. はじめに

本書は、T&D 製データロガー(TR-7wf/nw)に対しデータの書込み/読出しを行う CAO プロバイダのユーザーズガイドです。本書で扱う CAO プロバイダ(CaoProvTandDTR7nw.dll)を TR-7wf/nw プロバイダと呼びます。

第 2 章に TR-7wf/nw プロバイダの概要、変数やコマンドの詳細を記載しています。

TR-7wf/nw プロバイダで実装している通信コマンドは、通信先となるデータロガー(TR-7wf/nw)に依存します。通信の詳細については TR-7wf/nw シリーズの「TR-7wf/7nw HTTP 通信仕様書(Ver0.91)」を参照してください。

2. プロバイダの概要

2.1. 概要

TR-7wf/nw プロバイダは HTTP サーバーとして動作します. T&D 製データロガーからの定期的な POST に応答し, データの書き込み/読出しを行います.

本プロバイダの HTTP サーバー機能を使用するためには ACL 設定や Firewall 設定が必要です.

また, 下図 2-1 使用イメージが本プロバイダとデバイスの使用時のイメージになります.

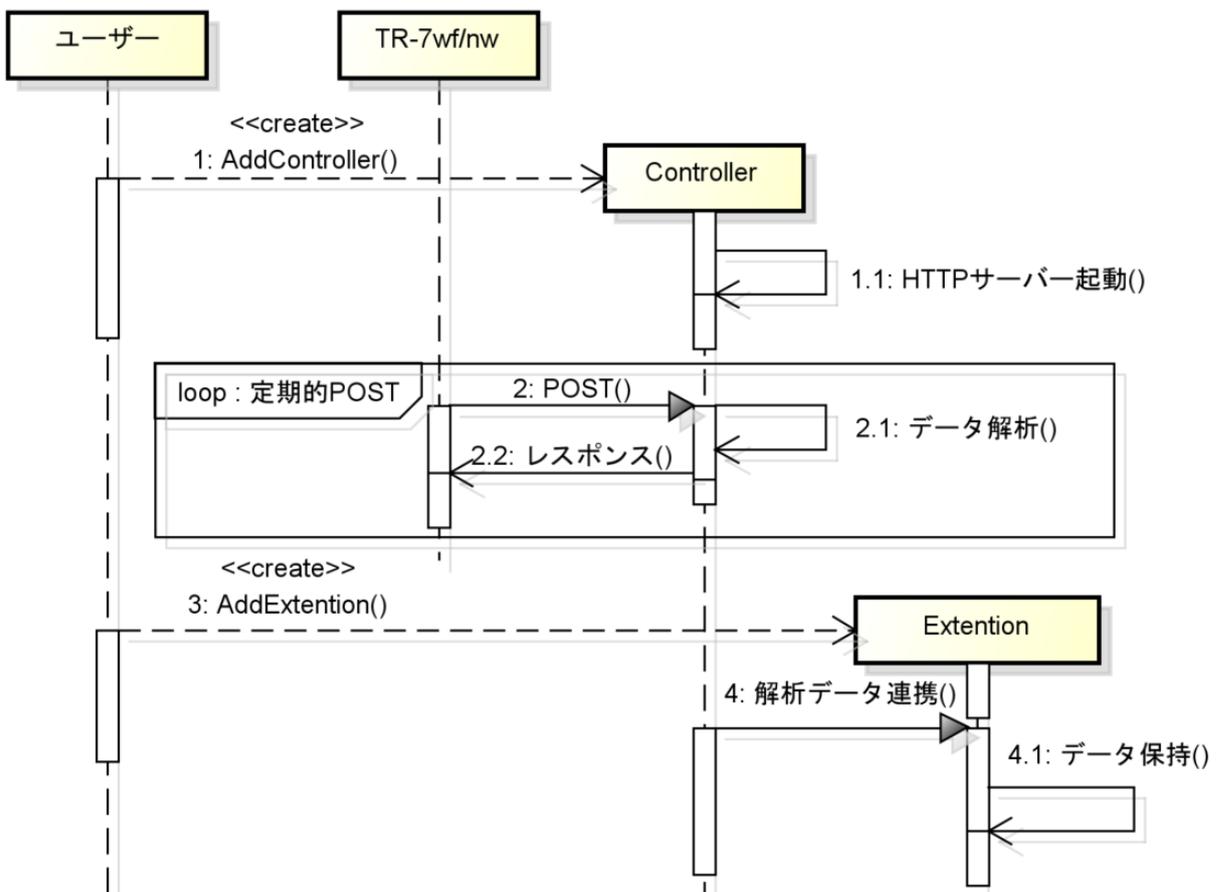


図 2-1 使用イメージ

- CaoWorkspace::AddController メソッドを実行すると, HTTP サーバーが起動します.
- データは参照する TR-7nfnw 毎に CaoExtention で管理します. キーとなる IP アドレスを指定して CaoController::AddExtention メソッドで CaoExtention のインスタンスを生成してください.
- CaoExtention::AddVariable メソッドで変数を生成することで, 各データの書き込み/読出しが可能になります.

TR-7wf/nw プロバイダのファイル形式は DLL(Dynamic Link Library)であり, CAO エンジンから使用時に動的ロードされます. TR-7wf/nw プロバイダを使用するにあたっては ORiN2SDK をインストールするか, 下表を参照して手作業でレジストリ登録を行う必要があります.

表 2-1 TR-7wf/nw プロバイダ

ファイル名	CaoProvTandDTR7nw.dll
ProgID	CaoProv.TandD.TR7nw
レジストリ登録 ¹	RegistAsm.bat CaoProvTandDTR7nw.dll
レジストリ登録の抹消	UnregistAsm.bat CaoProvTandDTR7nw.dll

¹ バッチファイル(RegistAsm.bat や UnregistAsm.bat)は, {ORiN2 インストールフォルダ}\DotNet\BAT 以下に存在します.

2.2. メソッド・プロパティ

2.2.1. CaoWorkspace::AddController メソッド

TR-7wf/nw プロバイダは AddController 時に HTTP サーバー用の宛先 URI を指定し、HTTP サーバー機能をリスン状態にします。



AddController (<bstrCtrlName:BSTR>,<bstrProvName:BSTR>,
<bstrPcName:BSTR>,<[bstrOption:BSTR]>)

<bstrCtrlName> : [in] コントローラ名
 <bstrProvName> : [in] プロバイダ名. 固定値 =” CaoProv.TandD.TR7nw”
 <bstrPcName> : [in] プロバイダの実行マシン名 (未使用)
 <bstrOption> : [in] オプション文字列

以下にオプション文字列に指定するリストを示します。

表 2-2 CaoWorkspace::AddController のオプション文字列

オプション	意味
DEST_URI =<HTTP サーバー宛先 URI>	必須. HTTP サーバーの宛先 URI を指定します. (参照:補足)
TYPE [=<言語タイプ>]	言語タイプを指定します. 0 : JP (デフォルト値) 1 : EU 2 : US
ON_MESSAGE [=<TRUE/FALSE>]	POST 応答時のメッセージの有無を指定します. (デフォルト:TRUE)

※ACL 設定や Firewall 設定をせずに本メソッドを実行すると、エラーコード: 0x80004005: を返すことがあります。



```
var option = "DEST_URI=http://192.168.3.42:80/,TYPE=0,ON_MESSAGE=TRUE";
this.ctrl = this.ws.AddController("Sample","CaoProv.TandD.TR7nw","",option);
```

2.2.2. GaoController::GetVariableNames プロパティ

2.3 に示しているシステム変数名の一覧を取得します。

使用例

```
var varNames = this.ctrl.get_VariableNames();
```

2.2.3. GaoController::AddVariable メソッド

HTTP サーバー機能の受信 IP アドレスと拒否 IP アドレスの書き込み/読出しを行うための変数オブジェクトを作成します。

書式

AddVariable (<bstrVariableName:BSTR>, [<bstrOption:BSTR>])

<bstrVariableName > : [in] 変数名

<bstrOption> : [in] オプション文字列(今回は使用しません)

使用例

```
var clientIP = this.ctrl.AddVariable("@CLIENT_IP", null);
```

2.2.4. GaoController::Execute メソッド

GaoController クラスの Execute メソッドは、コマンドを実行するためのメソッドです。各コマンドの詳細はコマンドリファレンスを参照してください。

書式

Execute (<bstrCommandName:VT_BSTR>,[<vntParam : VT_VARIANT>])

<bstrCommandName> : [in] コマンド名

<vntParam> : [in] パラメータ

使用例

```
var result = this.ctrl.Execute("Stop", null);
```

2.2.5. CaoController::AddExtention メソッド

CaoController クラスの AddExtention メソッドは, CaoExtention オブジェクトを作成するためのメソッドです. メソッド引数に TR-7wf/nw の IP アドレスを指定します.

書式 AddExtention(<bstrExtentionName:BSTR>, [<bstrOption:BSTR>])

<bstrExtentionName > : [in] 変数名
<bstrOption> : [in] オプション文字列

以下にオプション文字列に指定するリストを示します.

表 2-3 CaoController::AddExtention のオプション文字列

オプション	意味
IP =< IP アドレス>	必須. TR-7wf/7nw の IP アドレスを指定します.

使用例

```
var extOption = "IP=192.168.3.81";  
this.extention = this.ctrl.AddExtension("Sample", extOption);
```

2.2.6. CaoController::OnMessage イベント

TR-7wf/nw プロバイダは、CaoWorkspace::AddController メソッドの ON_MESSAGE オプションを TRUE にした場合、HTTP サーバーの応答時に、CaoController クラスの OnMessage イベントとしてクライアントに定型メッセージを受け渡します。このとき、Message::Value プロパティに定型メッセージを格納します。

HTTP サーバーが応答したクライアントの IP アドレスや応答タイミングなどを確認する際にご使用ください。

表 2-4 Message::Number プロパティの値とデータ種別の対応

Number プロパティ	データ種別	データ型
1	定型メッセージ	BSTR

定型メッセージのフォーマットは、下記です。

IP:{POSTしてきた IP アドレス},Command:{応答コマンド番号} finish.

応答コマンド番号について以下のリストに示します。

表 2-5 応答コマンド番号

番号	意味
0	HTTP サーバーは、TR-7wf/nw からの エラー を応答しました。
1	HTTP サーバーは、TR-7wf/nw からの 時刻データの要求 を応答しました。
2	HTTP サーバーは、TR-7wf/nw からの 動作設定値の要求 を応答しました。 ※動作設定値送信用のデータが送信されます。
3	HTTP サーバーは、TR-7wf/nw からの 記録データ送信 を応答しました。 ※現在値データにデータが格納されます。
4	HTTP サーバーは、TR-7wf/nw からの 動作設定値送信 を応答しました。 ※動作設定値受信用にデータが格納されます。
5	HTTP サーバーは、TR-7wf/nw からの 警報送信 を応答しました。 ※警報データにデータが格納されます。

2.2.7. GaoExtention::AddVariable メソッド

TR-7wf/7nw のデータについて書込み/読出しを行うための変数オブジェクトを作成します。

書式

AddVariable (<bstrVariableName:BSTR>, [<bstrOption:BSTR>])

<bstrVariableName> : [in] 変数名

<bstrOption> : [in] オプション文字列(今回は使用しません)

使用例

```
this.value = this.extention.AddVariable("@FLG_VALUE");
```

2.2.8. GaoVariable::get_Value プロパティ

2.3 に示しているシステム変数に対応した値を取得します。

使用例

```
var val = (int)this.value.Value;
```

2.2.9. GaoVariable::put_Value プロパティ

2.3 に示しているシステム変数に対応した値を設定します。

使用例

```
int setValue = 0;  
this.value.Value = tempValue;
```

2.3. 変数一覧

2.3.1. CaoController クラス

表 2-6 CaoController クラス システム変数一覧

変数名	データ型	説明	属性	
			get	put
@CLIENT_IP	ARRAY BSTR	本プロバイダに POST してきたクライアント IP の一覧を配列返す.	○	-
@DENY_IP	ARRAY BSTR	本プロバイダにて応答を返さない IP アドレスの一覧を配列で返す.	○	○

2.3.2. CaoExtention クラス

2.3.2.1. 動作設定値送信用

TR-7wf/nw プロバイダ(HTTP サーバー)から TR-7wf/nw への動作設定値送信用の変数.

@FLG_VALUE を 1 に設定することで,

TR-7wf/nw に@FLG_VALUE を除く動作設定値用変数の値の取得要求を送信します.

取得要求の送信後, @FLG_VALUE は 0 に自動で更新します.

表 2-7 CaoExtention クラス システム変数一覧 (動作設定値送信用)

変数名	データ型	説明	属性	
			get	put
@FLG_VALUE	I4	POST 応答時の動作設定情報の Flag 値 この値を変更することにより, 下記が可能 0 : 何もしない 1 : 動作設定値の取得要求 (サーバー → TR-7wf/nw) TR-7wf/nw に@FLG_VALUE 以外の 動作設定値送信用の変数値を 動作設定値として読み込ませる 2 : 動作設定値送信要求 (TR-7wf/nw → サーバー) TR-7wf/nw の動作設定値を 動作設定値受信用の変数に読み込 む デフォルト値 ※1 or 2 の応答が完了後, 値は自動で 0 に戻 ります.	○	○
@SET_TIME_DIFF	I2	時差(分) -720~720	○	○
@SET_SUMMER_TIME_FLG	BOOL	サマータイム FALSE : OFF TRUE : ON	○	○
@SET_SUMMER_TIME_DIFF	I2	サマータイム補正值(分) デフォルト:60分	○	○

変数名	データ型	説明	属性	
			get	put
@SET_DEVICE_NAME	BSTR	機器名称	○	○
@SET_GROUP_NAME	BSTR	グループ名	○	○
@SET_CH1_NAME	BSTR	Ch.1 名称	○	○
@SET_CH2_NAME	BSTR	Ch.2 名称	○	○
@SET_VIEW_UNIT	UI1	表示単位 0 : 摂氏(°C) 1 : 華氏(°F)	○	○
@SET_CH1_LOW_ENABLE	BOOL	Ch.1 下限 警報監視の有無 FALSE : OFF TRUE : ON	○	○
@SET_CH1_HIGHT_ENABLE	BOOL	Ch.1 上限 警報監視の有無 FALSE : OFF TRUE : ON	○	○
@SET_CH2_LOW_ENABLE	BOOL	Ch.2 下限 警報監視の有無 FALSE : OFF TRUE : ON	○	○
@SET_CH2_HIGHT_ENABLE	BOOL	Ch.2 上限 警報監視の有無 FALSE : OFF TRUE : ON	○	○
@SET_CH1_SENSOR_ENABLE	BOOL	Ch.1 センサー 警報監視の有無 FALSE : OFF TRUE : ON	○	○
@SET_CH2_SENSOR_ENABLE	BOOL	Ch.2 センサー 警報監視の有無 FALSE : OFF TRUE : ON	○	○
@SET_CH1_UPPER_LIMIT_VALUE	R4	Ch.1 上限値 設定温度 ※単位は, @SET_CH1_ATTRIBUTE を参照	○	○
@SET_CH1_LOW_VALUE	R4	Ch.1 下限値 設定温度 ※単位は, @SET_CH1_ATTRIBUTE を参照	○	○
@SET_CH1_JUDGE_TIME	UI2	Ch.1 判定時間(秒) 30~65520 60 単位	○	○

変数名	データ型	説明	属性	
			get	put
@SET_CH2_UPPER_LIMIT_VALUE	R4	Ch.2 上限値 設定温度 or 設定湿度 ※単位は, @SET_CH2_ATTRIBUTE を参照	○	○
@SET_CH2_LOW_VALUE	R4	Ch.2 下限値 設定温度 or 設定湿度 ※単位は, @SET_CH2_ATTRIBUTE を参照	○	○
@SET_CH2_JUDGE_TIME	UI2	Ch.2 判定時間(秒) 30~65520 60 単位	○	○
@SET_SEND_SAPN	UI4	自動送信間隔(秒) 60~86400	○	○
@SET_RECORD_SPAN	UI2	記録間隔(秒) 1~3600	○	○
@SET_CH1_ATTRIBUTE	UI1	Ch.1 記録属性 0x0D : 温度(°C) 0x0E : 温度(F)	○	○
@SET_CH2_ATTRIBUTE	UI1	Ch.2 記録属性 0x0D : 温度(°C) 0x0E : 温度(F) 0xD0 : 湿度 0xD1 : 高精度湿度 ※TR-71wf: 0x0D or 0x0E TR-72wf: 0xD0 or 0xD1	○	○
@SET_RECORD_MODE	UI1	記録モード 0x00 : エンドレス 0x80 : ワンタイム	○	○

2.3.2.2. 動作設定値受信用

TR-7wf/nw から受信した動作設定値格納用の変数.

@FLG_VALUE を 2 に設定することで,

TR-7nfnw に動作設定値送信要求を送信し, 動作設定値受信用の変数が最新の情報に更新されます.

取得要求の送信後, @FLG_VALUE は 0 に自動で更新します.

表 2-8 CaoExtetion クラス システム変数一覧(動作設定値受信用)

変数名	データ型	説明	属性	
			get	put
@GET_TIME_DIFF	I2	時差(分) -720~720	○	-
@GET_SUMMER_TIME_FLG	BOOL	サマータイム FALSE : OFF TRUE : ON	○	-
@GET_SUMMER_TIME_DIFF	I2	サマータイム補正值(分) デフォルト:60分	○	-
@GET_DEVICE_NAME	BSTR	機器名称	○	-
@GET_GROUP_NAME	BSTR	グループ名	○	-
@GET_CH1_NAME	BSTR	Ch.1 名称	○	-
@GET_CH2_NAME	BSTR	Ch.2 名称	○	-
@GET_CH1_LOW_ENABLE	BOOL	Ch.1 下限 警報監視の有無 FALSE : OFF TRUE : ON	○	-
@GET_CH1_HIGHT_ENABLE	BOOL	Ch.1 上限 警報監視の有無 FALSE : OFF TRUE : ON	○	-
@GET_CH2_LOW_ENABLE	BOOL	Ch.2 下限 警報監視の有無 FALSE : OFF TRUE : ON	○	-
@GET_CH2_HIGHT_ENABLE	BOOL	Ch.2 上限 警報監視の有無 FALSE : OFF TRUE : ON	○	-

変数名	データ型	説明	属性	
			get	put
@GET_CH1_SENSOR_ENABLE	BOOL	Ch.1 センサー 警報監視の有無 FALSE : OFF TRUE : ON	○	-
@GET_CH2_SENSOR_ENABLE	BOOL	Ch.2 センサー 警報監視の有無 FALSE : OFF TRUE : ON	○	-
@GET_CH1_UPPER_LIMIT_VALUE	R4	Ch.1 上限値 (記録属性 0x0D, 0x0E 共通) 設定温度 ※単位は, @GET_CH1_ATTRIBUTE を参照	○	-
@GET_CH1_LOW_VALUE	R4	Ch.1 下限値 (記録属性 0x0D, 0x0E 共通) 設定温度 ※単位は, @GET_CH1_ATTRIBUTE を参照	○	-
@GET_CH1_JUDGE_TIME	UI2	Ch.1 判定時間(秒) 30~65520 60 単位	○	-
@GET_CH2_UPPER_LIMIT_VALUE	R4	Ch.2 上限値 2UI1 設定温度 or 設定湿度 ※単位は, @GET_CH2_ATTRIBUTE を参照	○	-
@GET_CH2_LOW_VALUE	R4	Ch.2 下限値 2UI1 設定温度 or 設定湿度 ※単位は, @GET_CH2_ATTRIBUTE を参照	○	-
@GET_CH2_JUDGE_TIME	UI2	Ch.2 判定時間(秒) 30~65520 60 単位	○	-
@GET_SEND_SPAN	UI4	自動送信間隔(秒) 60~86400	○	-
@GET_RECORD_SPAN	UI2	記録間隔(秒) 1~3600	○	-
@GET_CH1_ATTRIBUTE	UI1	Ch.1 記録属性 0x0D : 温度(°C) 0x0E : 温度(F)	○	-
@GET_CH2_ATTRIBUTE	UI1	Ch.2 記録属性 0x0D : 温度(°C)	○	-

変数名	データ型	説明	属性	
			get	put
		0x0E : 温度(F) 0xD0 : 湿度 0xD1 : 高精度湿度 ※TR-71wf: 0x0D or 0x0E TR-72wf: 0xD0 or 0xD1		
@GET_RECORD_MODE	UI1	記録モード 0x00 : エンドレス 0x80 : ワンタイム	○	-
@SERIAL_NO	UI4	シリアル番号 4UI1 JP TR-71wf: 52120000～3FFFF TR-72wf: 52140000～5FFFF EU TR-71wf: 42120000～3FFFF TR-72wf: 42140000～5FFFF US TR-71wf: 32120000～3FFFF TR-72wf: 32140000～5FFFF JP TR-71nw: 52160000～3FFFF TR-72nw: 52180000～3FFFF 他 TR-71nw: 62160000～3FFFF TR-72nw: 62180000～3FFFF	○	-
@DEVICE_VER	I2	ファームウェアバージョン 0～9999 値＝バージョン-100 例:214→Ver2.14	○	-
@DEVICE_POWER	UI1	電池残量(電池レベル) 0～5	○	-
@RECORD_STATUS	UI1	記録状態 0x00 : 停止中 0x01 : 記録中 0x02 : 予約開始待ち	○	-
@CH1_SLOPE	I2	Ch.1 傾き 0～65535 値＝傾き-1000 例:1100→傾きが 1.1	○	-
@CH1_SEGMENT	I2	Ch.1 切片 0～65535 値＝切片-1000 例:2400→切片が 2.4	○	-
@CH2_SLOPE	I2	Ch.2 傾き 0～65535 値＝傾き-1000 例:1600→傾きが 1.6	○	-

変数名	データ型	説明	属性	
			get	put
@CH2_SEGMENT	I2	Ch.2 切片 0~65535 値=切片-1000 例:3000→切片が 3.0	○	-
@RECORD_START_STAUS	UI1	記録開始状態 0x00 : 即時 0x01 : 予約 0x02 : マニュアル	○	-
@AUTO_SEND_PRESENCE	UI1	指定間隔での自動送信の有無 0 : 無効 1 : 有効	○	-
@REGISTER_CODE	UI4	登録コード 0x00000000~0x99999999	○	-
@BACKUP_STATUS	UI1	記録状態のバックアップ(TR-7wf Ver.104 以降)	○	-
@MAC_ADDRESS	ARRAY UI1	MAC アドレス(TR-7nw のみ)	○	-

2.3.2.3. 現在値データ

TR-7wf/nw から定期的に送信されてくる現在値データの格納用変数

表 2-9 CaoExtetion クラス システム変数一覧(現在値データ)

変数名	データ型	説明	属性	
			get	put
@CURRENT_DATA_SERIAL	BSTR	親機のシリアル番号	○	-
@CURRENT_DATA_MODEL	BSTR	親機の型番	○	-
@CURRENT_DATA_NAME	BSTR	親機名	○	-
@CURRENT_DATA_TIME_DIFF	I4	UTC のオフセット(単位:分) UTC-ローカルタイムの変換式 local_time = UTC + time_diff + std_bias(標準時間中) local_time = UTC + time_diff + dst_bias(夏時間中) 例) 日本:540 / 太平洋標準時:-480	○	-
@CURRENT_DATA_STD_BIAS	I4	標準時間のオフセット(分)	○	-
@CURRENT_DATA_DST_BIAS	I4	夏時間のオフセット(単位:分)	○	-
@CURRENT_DATA_TIME_ZONE	BSTR	タイムゾーンを表す文字列	○	-
@CURRENT_DATA_GROUP_NUMBER	I4	グループ番号	○	-
@CURRENT_DATA_GROUP_NAME	BSTR	グループ名	○	-
@CURRENT_DATA_REMORT_SERIAL	BSTR	子機のシリアル番号	○	-
@CURRENT_DATA_REMORT_MODEL	BSTR	子機の型番	○	-
@CURRENT_DATA_REMORT_NUMBER	I4	子機番号	○	-
@CURRENT_DATA_REMORT_NAME	BSTR	子機名	○	-

変数名	データ型	説明	属性	
			get	put
@CURRENT_DATA_REMORT_RSSI	BSTR	親機(中継機) - 子機間の電波強度 5: 強い ~ 1: 弱い, 0: 通信できない	○	-
@CURRENT_DATA_CH1_NUM	BSTR	Ch.1 チャンネル番号	○	-
@CURRENT_DATA_CH1_SCALE_EXPR	BSTR	Ch.1 スケール変換式 (参照:5.3)	○	-
@CURRENT_DATA_CH1_NAME	BSTR	Ch.1 チャンネル名	○	-
@CURRENT_DATA_CH1_UNIX_TIME	UI4	Ch.1 現在値の時刻 (世界協定時刻(UTC) 1970年1月1日からの経過秒数)	○	-
@CURRENT_DATA_CH1_TIME_STR	BSTR	Ch.1 現在値の時刻の文字列 (親機情報の時差情報を使って変換したローカルタイム)	○	-
@CURRENT_DATA_CH1_VALUE	BSTR	Ch.1 現在値 @CURRENT_DATA_CH1_SCALE_EXPR に値がある場合は、スケール変換した値になる @CURRENT_DATA_CH1_SCALE_EXPR に値がない場合は、 @CURRENT_DATA_CH1_UNIT で表した値になる	○	-
@CURRENT_DATA_CH1_UNIT	BSTR	Ch.1 現在値の単位 @CURRENT_DATA_CH1_SCALE_EXPR に値がある場合は、 スケール変換式で指定した単位になる	○	-
@CURRENT_DATA_CH1_BATT	BSTR	子機の電池残量(5:ある ~ 0:なし), -1:電池残量情報がない機器	○	-
@CURRENT_DATA_CH2_NUM	BSTR	Ch.2 チャンネル番号	○	-
@CURRENT_DATA_CH2_SCALE_EXPR	BSTR	Ch.2 スケール変換式 (参照:5.3)	○	-
@CURRENT_DATA_CH2_NAME	BSTR	Ch.2 チャンネル名	○	-
@CURRENT_DATA_CH2_UNIX_TIME	UI4	Ch.2 現在値の時刻 (世界協定時刻(UTC) 1970年1月1日からの経過秒数)	○	-

変数名	データ型	説明	属性	
			get	put
		経過秒数)		
@CURRENT_DATA_CH2_TIME_STR	BSTR	Ch.2 現在値の時刻の文字列 (親機情報の時差情報を使って変換したローカルタイム)	○	-
@CURRENT_DATA_CH2_VALUE	BSTR	Ch.2 現在値 @CURRENT_DATA_CH2_SCALE_EXPR に値があるときは、スケール変換した値になる @CURRENT_DATA_CH2_SCALE_EXPR に値がないときは、 @CURRENT_DATA_CH2_UNIT で表した値になる	○	-
@CURRENT_DATA_CH2_UNIT	BSTR	Ch.2 現在値の単位 @CURRENT_DATA_CH2_SCALE_EXPR に値がある場合は、 スケール変換式で指定した単位になる	○	-
@CURRENT_DATA_CH2_BATT	BSTR	Ch.2 子機の電池残量(5:ある ~ 0:なし), -1: 電池残量情報がない機器	○	-

2.3.2.4. 警報データ

TR-7wf/nw からの警報時に送信されてくる警報データの格納用変数

表 2-10 CaoExtetion クラス システム変数一覧(警報データ)

変数名	データ型	説明	属性	
			get	put
@CURRENT_DATA_CH2_BATT	BSTR	Ch.2 子機の電池残量(5:ある ~ 0:なし), -1: 電池残量情報がない機器	○	-
@WARNING_CREATED_TIME	I4	このファイルを作成した時刻 (世界協定時刻(UTC) 1970年1月1日からの経過秒数)	○	-
@WARNING_TIME_DIFF	I4	UTC のオフセット(単位:分) UTC-ローカルタイムの変換式 local_time = UTC + time_diff + std_bias(標準時間中) local_time = UTC + time_diff + dst_bias(夏時間中) 例) 日本:540 / 太平洋標準時:-480	○	-
@WARNING_STD_BIAS	I4	標準時間のオフセット(単位:分)	○	-
@WARNING_DST_BIAS	I4	夏時間のオフセット(単位:分)	○	-
@WARNING_DEVICE_SERIAL_NUM	BSTR	機器のシリアル番号	○	-
@WARNING_DEVICE_MODEL_NUM	BSTR	機器の型番	○	-
@WARNING_DEVICE_NAME	BSTR	機器名	○	-
@WARNING_CH1_NUM	I4	チャンネル番号	○	-
@WARNING_CH1_STATE	I4	Ch1. 警報状態 0 : 不明 1 : 正常(警報なし) 2 : 上限値を超えた警報が発生中 3 : 下限値を下回った警報が発生中 4 : センサエラー警報が発生中	○	-
@WARNING_CH1_NAME	BSTR	Ch1. チャンネル名	○	-

変数名	データ型	説明	属性	
			get	put
@WARNING_CH1_UNIX_TIME	I4	Ch1. 時刻 (世界協定時刻(UTC) 1970年1月1日からの経過秒数)	○	-
@WARNING_CH1_ID	I4	Ch1. 識別番号 警報が発生する毎に1増加(255を超えると0に戻る)	○	-
@WARNING_CH1_VALUE	BSTR	Ch1. 警報値	○	-
@WARNING_CH1_UNIT	BSTR	Ch1. 現在値の単位	○	-
@WARNING_CH1_UPPER_LIMIT	BSTR	Ch1. 上限値	○	-
@WARNING_CH1_LOWER_LIMIT	BSTR	Ch1. 下限値	○	-
@WARNING_CH2_NUM	I4	Ch2. チャンネル番号	○	-
@WARNING_CH2_STATE	I4	Ch2. 警報状態 0 : 不明 1 : 正常(警報なし) 2 : 上限値を超えた警報が発生中 3 : 下限値を下回った警報が発生中 4 : センサエラー警報が発生中	○	-
@WARNING_CH2_NAME	BSTR	Ch2. チャンネル名	○	-
@WARNING_CH2_UNIX_TIME	I4	Ch2. 時刻 (世界協定時刻(UTC) 1970年1月1日からの経過秒数)	○	-
@WARNING_CH2_ID	I4	Ch2. 識別番号 警報が発生するごとに1増加(255を超えると0に戻る)	○	-
@WARNING_CH2_VALUE	BSTR	Ch2. 警報値	○	-
@WARNING_CH2_UNIT	BSTR	Ch2. 現在値の単位	○	-
@WARNING_CH2_UPPER_LIMIT	BSTR	Ch2. 上限値	○	-

変数名	データ型	説明	属性	
			get	put
@WARNING_CH2_LOWER_LIM IT	BSTR	Ch2. 下限値	○	-

3. コマンドリファレンス

3.1. CaoController クラス

表 3-1 CaoController クラス コマンド一覧

コマンド	機能	ページ
HTTP サーバー操作		
Start	HTTP サーバーをスタートさせます.	27
Stop	HTTP サーバーをストップさせます.	28
Restart	HTTP サーバーを再スタートさせます.	28

3.1.1. CaoController::Execute (“Start”) コマンド

HTTP サーバーをスタートさせてリスン状態にします.

書式

Start(<DEST_URI >,[<TYPE>],[<ON_MESSAGE>])

DEST_URI : [in] 必須. HTTP サーバーの宛先 URI を指定します.
(VT_BSTR)

TYPE : 言語タイプを指定します.
0 : JP (デフォルト値)
1 : EU
2 : US

ON_MESSAGE : POST 応答時のメッセージの有無を指定します.
(デフォルト:TRUE)

戻り値 : [out] TRUE:メソッド成功 , FALSE:メソッド失敗
※HTTP サーバーがリスン状態時に実行すると FALSE が返ります.

使用例

```
var param = new string[] { "http://192.168.3.42:80/", "0", "TRUE" };
var result = this.ctrl.Execute("Start", param);
```

※DEST_URI に ACL 設定にて未設定な URI を指定した場合,

「アクセスが拒否されました」というメッセージのエラーがスローされます.

上記後は, ACL 設定にて登録済の URL を指定して本コマンドを実行するまで, Stop コマンド(3.1.2)と Restart コマンド(3.1.3)の実行結果の戻り値は, FALSE を返します.

3.1.2. GaoController::Execute ("Stop") コマンド

HTTP サーバーをストップさせて停止状態にします。

書式

Stop ()

戻り値 : [out] TRUE:メソッド成功 , FALSE:メソッド失敗

使用例

```
var result = this.ctrl.Execute("Stop", null);
```

3.1.3. GaoController::Execute ("Restart") コマンド

HTTP サーバーを直近でリスンしていた状態で再スタートし、リスン状態にします。

書式

Restart()

戻り値 : [out] TRUE:メソッド成功 , FALSE:メソッド失敗

※HTTP サーバーがリスン状態時に実行すると FALSE が返ります。

使用例

```
var result = this.ctrl.Execute("Restart", null);
```

4. サンプルプログラム

以下に CaoExtention のシステム変数"@FLG_VALUE"から I4 の値設定, 取得をする場合の C#のサンプルを示します.

List 4-1

```
... (略) ...
using ORiN2. interop. CAO;

namespace TandDSample
{
    public partial class Sample : Form
    {
        private CaoEngine engine;
        private CaoWorkspaces wss;
        private CaoWorkspace ws;
        private CaoControllers ctrls;
        private CaoController ctrl;
        private CaoVariable value;

        public Sample()
        {
            InitializeComponent();
        }

        private void Sample_Load(object sender, EventArgs e)
        {
            // Caoエンジンの生成
            this.engine = new CaoEngine();
            this.wss = this.engine.Workspaces;
            this.ws = this.wss.Item(0);
            this.ctrls = this.ws.Controllers;

            // オプション文字列
            var option = "DEST_URI=http://192.168.3.42:80/, TYPE=0, ON_MESSAGE=TRUE";

            // TR-7wf/nw のControllerを追加
            this.ctrl = this.ws.AddController("Sample",
                                             "CaoProv. TandD. TR7nw",
                                             "",
                                             option);

            // Extentionを追加
            var extOption = "IP=192.168.3.81";
            this.extention =this.ctrl.AddExtension("test", extOption);

            // アクセス用変数を追加
            this.value =this.extention.AddVariable("@FLG_VALUE");
        }

        private void btnPut_Click(object sender, EventArgs e)
        {
            // 値の設定
            this.value.Value = int.Parse(txtItemValue.Text);
        }

        private void btnGet_Click(object sender, EventArgs e)
        {
            // 値の取得
            Int val = (int) this.value.Value;
        }
    }
}
```

```
        txtItemValue.Text = val.ToBSTR();
    }

private void Sample_FormClosing(object sender, FormClosingEventArgs e)
{
    // Cao関連オブジェクトの開放
    this.ctrls.Clear();
    if (this.ctrl != null)
    {
        System.Runtime.InteropServices.Marshal.ReleaseComObject(this.ctrl);
        this.ctrl = null;
    }

    this.wss.Clear();

    if (this.ctrls != null)
    {
        System.Runtime.InteropServices.Marshal.ReleaseComObject(this.ctrls);
        this.ctrls = null;
    }

    if (this.ws != null)
    {
        System.Runtime.InteropServices.Marshal.ReleaseComObject(this.ws);
        this.ws = null;
    }

    if (this.wss != null)
    {
        System.Runtime.InteropServices.Marshal.ReleaseComObject(this.wss);
        this.wss = null;
    }

    if (this.engine != null)
    {
        System.Runtime.InteropServices.Marshal.ReleaseComObject(this.engine);
        this.engine = null;
    }
}
}
```

5. 補足

5.1. ACL 設定

HTTP サーバー機能を使用するために、宛先 URI の ACL 設定が必要です。

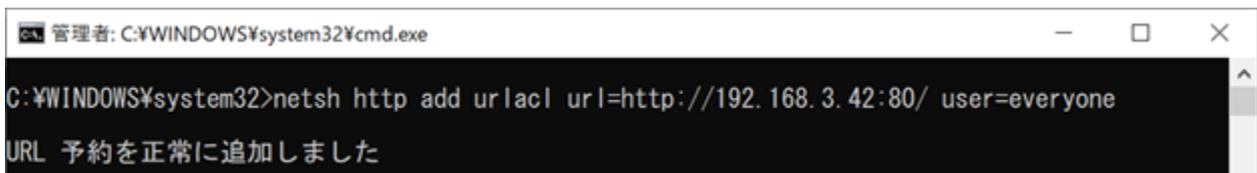
5.1.1. 追加

HTTP サーバー機能を使用するために、宛先 URI の ACL 設定の追加が必要です。
コマンドプロンプトを管理者権限で起動し、下記コマンドを実行してください。

```
netsh http add urlacl url=http://{IP アドレス or ホスト名}:{ポート番号}/ user=everyone
```

※上記 url に指定する内容は、

CaoWorkspace::AddController メソッドの DEST_URI オプションに指定するものです。



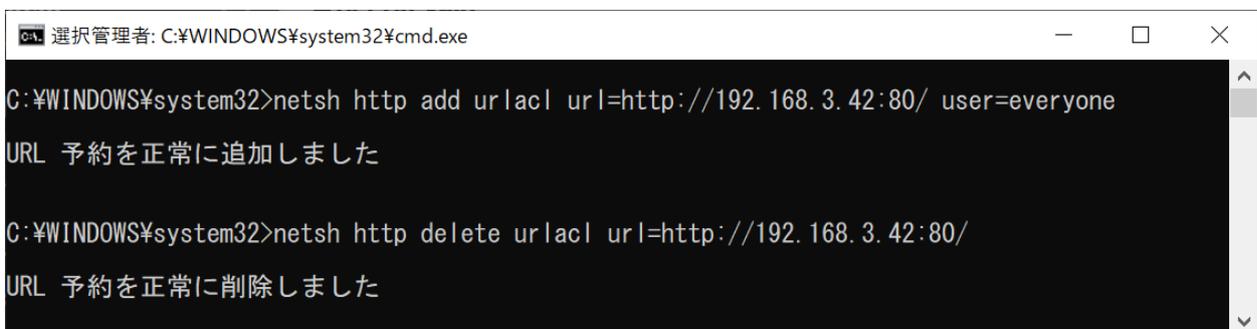
```
管理者: C:\WINDOWS\system32\cmd.exe
C:\WINDOWS\system32>netsh http add urlacl url=http://192.168.3.42:80/ user=everyone
URL 予約を正常に追加しました
```

図 5-1 ACL 設定(追加)

5.1.2. 削除

誤った ACL 設定を追加した場合など、ACL 設定の削除を行う場合には、
コマンドプロンプトを管理者権限で起動し、下記コマンドを実行してください。

```
netsh http delete urlacl url=http://{IP アドレス or ホスト名}:{ポート番号}/
```



```
選択管理者: C:\WINDOWS\system32\cmd.exe
C:\WINDOWS\system32>netsh http add urlacl url=http://192.168.3.42:80/ user=everyone
URL 予約を正常に追加しました

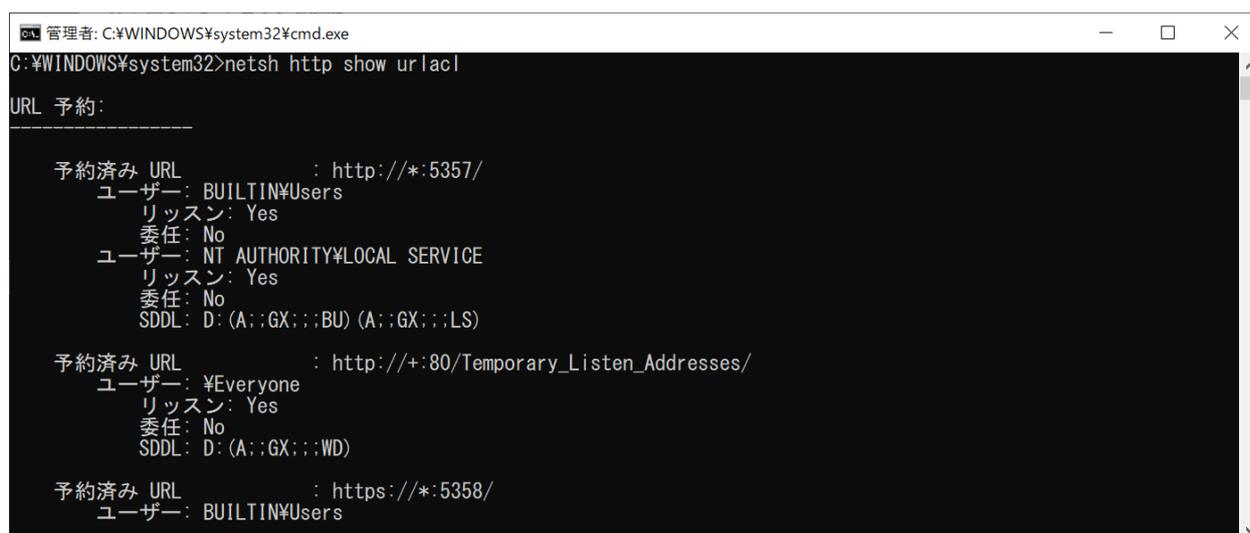
C:\WINDOWS\system32>netsh http delete urlacl url=http://192.168.3.42:80/
URL 予約を正常に削除しました
```

図 5-2 ACL 設定(削除)

5.1.3. 確認

ACL 設定の確認を行う場合には、
コマンドプロンプトを管理者権限で起動し、下記コマンドを実行してください。

```
netsh http show urlacl
```



```
管理者: C:\WINDOWS\system32\cmd.exe
C:\WINDOWS\system32>netsh http show urlacl

URL 予約:
-----
予約済み URL      : http://*:5357/
ユーザー: BUILTIN\Users
リッスン: Yes
委任: No
ユーザー: NT AUTHORITY\LOCAL SERVICE
リッスン: Yes
委任: No
SDDL: D: (A;;GX;;;BU) (A;;GX;;;LS)

予約済み URL      : http://+:80/Temporary_Listen_Addresses/
ユーザー: \Everyone
リッスン: Yes
委任: No
SDDL: D: (A;;GX;;;WD)

予約済み URL      : https://*:5358/
ユーザー: BUILTIN\Users
```

図 5-3 ACL 設定(確認)

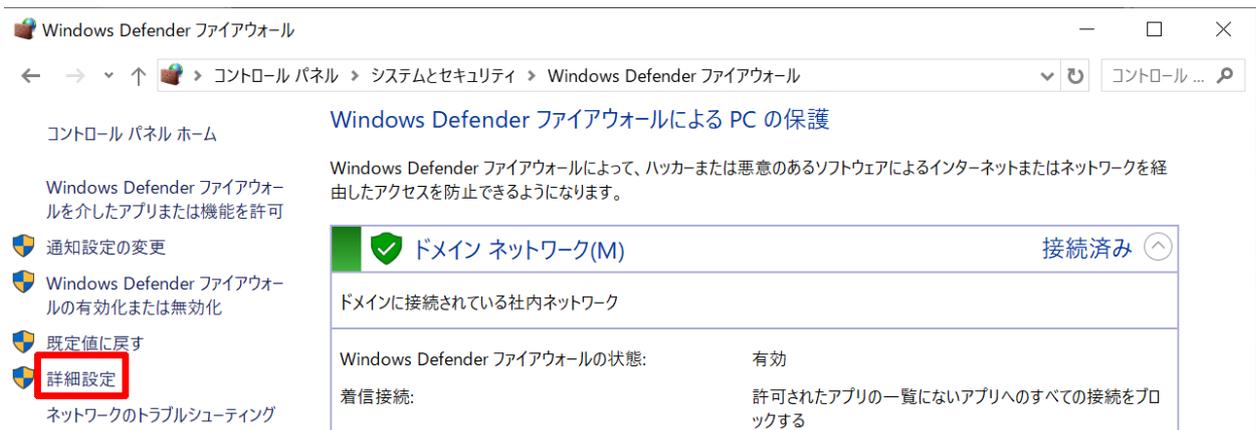
5.2. Firewall 設定

HTTP サーバー機能を使用するために、ACL 設定にて指定したポート番号の受信と送信の許可が必要です。

Windows の場合の設定例を示します。

エクスプローラーにて、

コントロールパネル > システムとセキュリティ > Windows Defender ファイアウォール を表示し、詳細設定をクリック



受信の規則をクリック後、新しい規則をクリック

※送信の規則も以下手順は同様です。



ポートを選択し、次へをクリック



ACL 設定にて指定したポート番号を入力し、次へをクリック



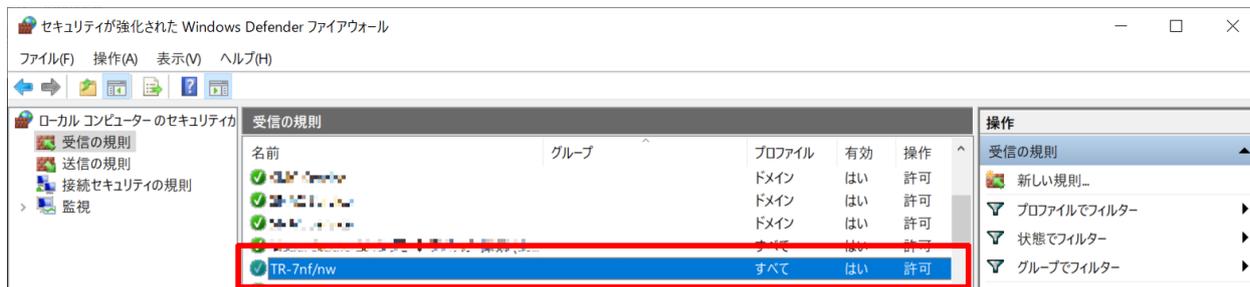
必要な適用範囲にチェックをつけ、次へをクリック



名前や説明を入力し、完了をクリック



受信の規則に追加した規則が表示され有効になっていることを確認



5.3. スケール変換式

本項は、現在値データの@CURRENT_DATA_CH1_SCALE_EXPR と @CURRENT_DATA_CH2_SCALE_EXPR について説明します。

スケール変換式とは、測定値を別の単位で表示するときの変換式です。
スケール変換式の値は、下記のルールで記述します。

1. 改行 (0x0A) を区切り文字とする。以降、改行を¥n と表記する。
2. 改行区切りで、傾き,切片,有効数字,単位 を記述する。
3. スケール変換式の値には、必ず、改行が4文字含まれる。
4. スケール変換をしないときは、"¥n¥n¥n¥n" とする。
5. 有効数字が 0 のときは、有効数字に関する処理を行わないで記述する。

(例)

傾き 3.368
切片 1.285
有効数字 4
単位 kgf/cm2

3.368¥n1.285¥n4¥nkgf/cm2¥n

(例)

スケール変換しない
¥n¥n¥n¥n