

INDUSTRIAL  
EDGE  
SOLUTION  
WITH  
HARD REALTIME  
CAPABILITIES

**RT-edge**

Micronet.Co,

**マイクロネット**  
**Micronet**

INDUSTRIAL REALTIME EDGE COMPUTERS




# HMI コンテナ

RT-edge Software HMI Container  
ユーザーズマニュアル

 **株式会社マイクロネット**  
<http://www.mnc.co.jp>

TEL: +81(0)299-90-1733

本書で使用するマークについて

	ノート: 操作方法や手順等の補足情報や注釈を説明しています。
	情報: 製品を利用する上で有効な豆知識となる説明をしています。
	警告: 製品仕様上注意が必要な事象について説明しています。

Windows、Visual Studio は、米国 Microsoft Corporation の米国およびその他の国における商標または登録商標です。  
INtime は、米国 TenAsys Corporation の登録商標です。

TenAsys®, INtime®, eVM® and iRMX® are registered trademarks in USA of the TenAsys Corporation.

その他、本書に記載されている会社名、商品名は、各社の商標または登録商標です。

本書の内容を無断で転載することは禁止されています。

本書の内容に関しては、予告なしに変更することがあります。あらかじめご了承ください。

## 目次

用語解説	4
関連資料	6
1. 概要	7
1.1. RT-edge とサービスコンテナ .....	7
1.2. HMI コンテナ .....	9
2. 仕様	11
2.1. 動作環境 .....	11
2.2. 使用可能タグ数 .....	11
2.3. 画面仕様/ページ .....	11
2.4. タイル .....	12
3. コンテナ導入フロー	13
4. インストール	14
4.1. ファイル .....	14
4.2. ファイルのインストール .....	15
4.3. IIS の設定 .....	15
4.4. HMI 画面の作成 .....	17
4.5. 起動設定 .....	18
4.6. 動作確認 .....	19
4.7. 終了設定 .....	21
5. 設定概要	22
5.1. ECI 設定 .....	22
6. コンテナプロパティタグ	25
6.1. 一覧 .....	25
6.2. サービス動作自動スタート指定(.AutoRun) .....	25
7. HMI アプリケーション設定/ページ設定	26
7.1. アプリケーション設定 .....	26
7.2. ページ設定 .....	29
8. タイル一覧/タイル設定値	31
8.1. ラベル .....	31
8.2. ボタン .....	33
8.3. ランプ .....	49
8.4. チャート .....	51
8.5. ログ表示 .....	56
8.6. インナーフレーム .....	59
8.7. テキストボックス .....	60
9. 設定サンプル	62
10. 動作確認	64
10.1. 動作確認手順 .....	64

10.2. サービスインジケータタグ .....	65
10.3. RT-edge タグデータの妥当性について .....	65
<b>11. カスタムタイルの作成</b> .....	<b>66</b>
11.1. 開発環境 .....	66
11.2. 作成から運用までの流れ .....	66
11.3. タイル作成例(電源ボタン作成) .....	67
<b>12. 付録</b> .....	<b>76</b>
12.1. トラブルシューティング .....	76

## 用語解説

本ドキュメントにおいて使用される用語・略称について説明します:

表 1. 用語集

用語	説明
RT-edge	エッジコンピューティングを軸とする IT の情報処理と、FA における装置・機器の制御を融合し、密度の高い高頻度データ利用を可能とするソフトウェアプラットフォームです。 FA で要求されるハードリアルタイム制御を組み込むことで、情報処理と機器・装置制御を可能とするエッジコントローラを構成することができます。
RT-edge 基本ソフトウェア	RT-edge 機能の核となる機能・ライブラリを実装するパッケージソフトウェア製品です。
IoT ゲートウェイ	IoT において、端末とインターネットを介した遠隔サーバー(クラウド)がデータのやりとりをする際、中継する役割を担う機能。サーバーや送信経路であるインターネット負荷の軽減をします。
IT システム	オンプレミスもしくはクラウドを活用した業務システムやアプリケーション。
INtime	INtime for Windows: Windows と協調動作可能なリアルタイムカーネル拡張ソフトウェアです(RTOS ソフトウェア)。 INtime Distributed RTOS(dRTOS): Windows OS を必要とせず、スタンドアロンで動作するリアルタイム OS です。
RTA	RealTime Application: リアルタイムアプリケーションの略称。INtime 上で動作するローダブルプロセスの拡張子です。INtime 上で動作するローダブルアプリケーションは、RTA という拡張子を持ちます。
RSL	Realtime Shared Library: リアルタイム共有ライブラリの略称。INtime 上でアプリケーションがロード可能なライブラリです。Windows 上で使用される DLL(Dynamic Link Library)のようなものです。RTA から使用されるライブラリインタフェース等は、こちらを使用して作成することができます。
API	Application Programming Interface: アプリケーションプログラミングインタフェースの略称。RT-edge ではデバイスへのアクセスインタフェースとして API ライブラリを提供しています。
エッジアプリケーション	RT-edge 内コンテナにより集積されたデータ(RT-edge Object)を活用、処理実行するソフトウェアです。
エッジコンピューティング	RT-edge 内で稼働する制御コンテナソフトウェアにより装置・機器から収集した高密度なデータをリアルタイムに収集、分析、フィードバックします。IT システムとの情報連携。
オンプレミス	サーバーやソフトウェア等の情報システム、アプリケーション等のソフトウェアを管理する施設内に設置して運用すること。
クラウド	サーバーやストレージ等のインフラやソフトウェアを必要とせず、必要な IT リソースが、インターネットを通じてオンデマンドで得られる形態、サービス。
産業用 PC	高信頼性、耐環境性、長期供給等の特徴をもつ産業用途の PC。
データ収集	診断、分析を行う対象となるデータを集積する処理。
データ加工	集積されたデータを利用しやすい形に変更する処理。
サービス/EgService	RT-edge システムを構成する機能プロセス(rta/exe)です。
タグ/EgTag	瞬時値データ値 1 つを示すオブジェクトです。ユニーク名とグローバルなスコープを持ち、全ての EgService から読み書きが許されたオブジェクトです。タグは生成時にデータ型が確定され変更はできません。
リンクタグ	同一名称のタグを重複生成した場合に自動的に別名称で生成されるタグを指します。 通常のタグと同様、グローバルなスコープを持ち、全ての EgService から読み書きが許されたオブジェクトです。一つのタグに対し、異なるプロパティ情報を定義したい場合に使用します。
データセット/EgDataset	タグ 1 つ以上の組み合わせでデータ並び順(データ構造)を定義する名前付きオブジェクトです。
コレクタ/EgCollector	データセットに定義されたデータ構造に従って、同時刻のバイナリデータ列で生成し、データレコードとしてメールボックスに送信するオブジェクト (スレッド) です。

用語	説明
メールボックス/EgMailBox	時系列なデータセット、または時系列メッセージを FIFO で蓄えることができ、また受信イベントとして処理できるオブジェクトです。
タグ参照/TagRef	タグの参照として使用するオブジェクトです。タグの名前を保持し値は保持しません。サービスコンフィグファイルでデータセットの収集用タグとして定義することや、サービス内のオブジェクトとして定義することでサービスのメンバ変数として使用することができます。
コレクタ参照/CollectorRef	コレクタの参照として使用するオブジェクトです。コレクタの名前を保持しそれ以外のオブジェクトは保持しません。サービスコンフィグファイルでサービス内のオブジェクトとして定義することでサービス内のメンバ変数として使用することができます。
タグトリガー/TagTrigger	タグトリガーとは特定のタグの値が変化した場合に、サービス側でメッセージ通知を受けることができる機能です。サービスの ECI ファイルでタグトリガーとして登録したいタグ名を記載することで、タグ値の変更通知を受け取ることができます。
メッセージ	メールボックスで扱われる 1 レコード分のデータ、またはサービス間のコマンド、応答の電文です。
フレームワーク	フレームワークは、アプリケーションが API を組み合わせて実装するよくある処理についてマクロ化、自動化したものでサービスコンフィグファイルの記述により自動処理させることができます。
RT-edge コンテナ設定情報 (ECI)	RT-edge コンテナが RT-edge Object として展開する入出力データ定義の他、RT-edge コンテナフレームワークが、オブジェクト生成やコンテナサービス等自動処理するための定義設定情報(XML 型式)。
入力	RT-edge システムを中心に見た場合、外部の情報を RT-edge システムへ取り込む方向性のデータの流れを意味します。
出力	RT-edge システムを中心に見た場合、RT-edge システムが持つデータを外部に書き出す方向性のデータの流れを意味します。
RTCD	Realtime Common Data の略称。RT-edge システム上で最もベースとなる共有データ構造機能です。
RT-edge Object	RT-edge システム上で使用可能なオブジェクト群（機能群）の総称です。 例えば、センサーや装置から収集したデータをアプリケーション間で受け渡しを行う場合に使用するタグ、アプリケーション間でメッセージのやり取りを行う場合のメールボックス等、アプリケーション間でデータの受け渡しを行うケースにおいて利用されるオブジェクトです。 RT-edge Object は Windows アプリケーション間、INtime®アプリケーション間、Windows-INtime®アプリケーション間いずれの場合も利用可能です。
IIS	Internet Information Services の略。Microsoft が提供する Web サーバソフトウェアです。Web サイトやアプリケーションのホスティングを行い、HTTP、HTTPS、FTP などのプロトコルをサポートしています。セキュリティ、スケーラビリティ、管理ツールも充実しています。
タイル	画面上に表示されるボタンやランプといったコンポーネントをそれぞれタイルと呼びます。タイルを組み合わせることで画面の見た目を作成します。またタイルへ RT-edge のタグを割り当てることで、タグの値の入出力が可能となります。

## 関連資料

### RT-edge 製品に含まれる資料

表 2 .RT-edge 関連資料

名称	ファイル名	内容
RT-edge ユーザーズマニュアル	DOCRTEGEUSER.pdf	RT-edge システムの全般的な説明が記載されています。
RT-edge API リファレンス	DOCRTEGEAPI.pdf	RT-edge API の使用方法が記載されています。
RT-edge コンテナ作成マニュアル	DOCRTEGDSRV.pdf	RT-edge コンテナの構造、サンプルプロジェクトを利用した作成方法等について記載されています。

# 1. 概要

## 1.1. RT-edge とサービスコンテナ

RT-edge とは、エッジコンピューティングを軸とする IT の情報処理と、FA における装置・機器の制御を融合し、密度の高い高頻度データ利用を可能とするソフトウェアプラットフォームです。

RT-edge の利用により、装置やセンサーからの高密度なデータ収集、分析だけでなく、提供される開発ライブラリキットを使用し、タグデータをレジスタとした機器制御を行うハードリアルタイムエッジアプリケーションの開発が可能です。

### サービスコンテナ

RT-edge の処理ターゲットは、エッジコンピューティングを軸とした IT 情報処理(IT-Process)と、ミリ秒精度のハードリアルタイム性を要求される FA 制御(FA-Control)に分類され、ターゲットの機能に特化した専門処理サービスをコンテナ(サービスコンテナ)と呼びます。

IT 情報処理ターゲットは上位層にあり、主に外部システムからの要求指示の受付や、外部システムへのデータ公開、通信等を担う要素となります。IT 情報処理サービスコンテナは、制御システムのコンソール画面や外部システムから WEB ブラウザ経由でのアクセス機能、制御データ情報を外部クラウドストレージに保存する機能等、上位システムとの接続・インターフェースを提供します。

一方、FA 制御ターゲットは下位層に位置し、主に通信やハードウェアへの直接 I/O 入出力等により装置・機器制御を担う要素です。FA 制御サービスコンテナは、産業用フィールドバスやコントローラ通信プロトコルによるロボット制御、計測機器からのデータロギング、デジタルパルス出力等、装置・機器へのアクセスを提供します。

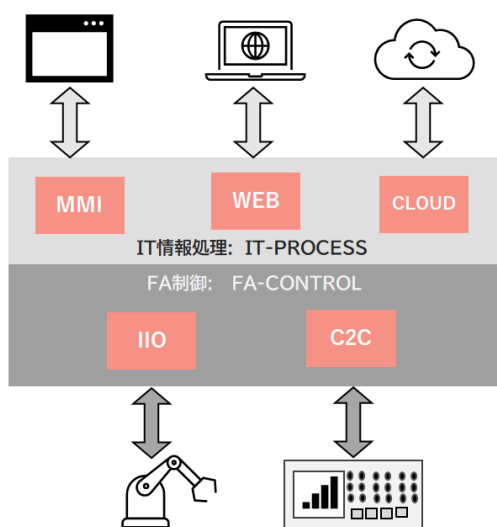


図 1. ターゲットとコンテナ

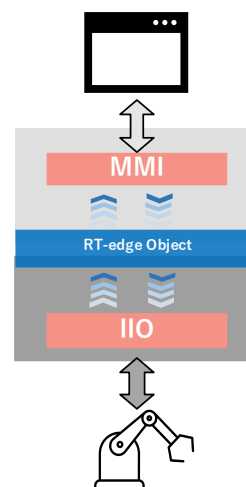


図 2. コンテナの役割





### サービスコンテナ例

WEB: IT 情報処理ターゲット内には、IIS(Internet Information Service)を介し、制御情報をインターネット上に公開する WEB サービスコンテナ  
 IIO: ロボットアーム制御に特化した産業 I/O サービスコンテナ

サービスコンテナはターゲットに特化した入出力データを RT-edge Object であるシステム内でグローバルにアクセス可能なタグ情報としてリンクし、このタグ情報のコレクションを公開します。

サービスコンテナは、タグ情報コレクションや、動作・挙動を決定するパラメータ設定と、ターゲット処理に特化した一つ以上の実行処理の集合体です:

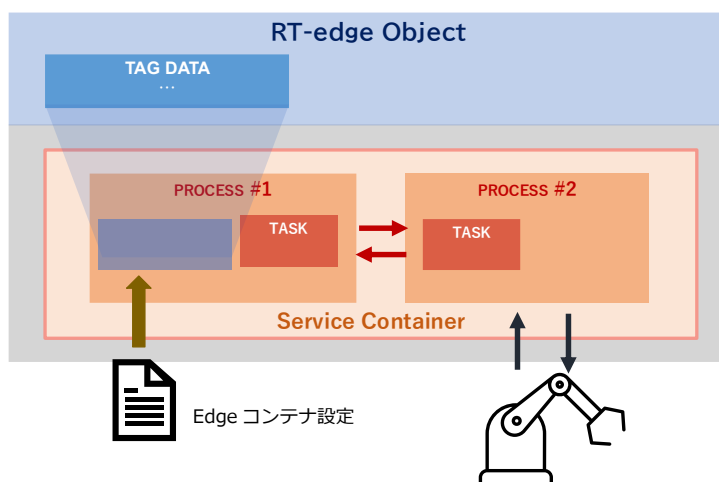


図 3. サービスコンテナの構造

### サービスコンテナ例

#### PROCESS #1

RT-edge Object を使用する処理プロセス (サービスハンドラ)

#### PROCESS #2

ターゲット特化処理プロセス

#### Edge コンテナ設定情報

タグデータコレクション等コンテナ設定

## 1.2. HMI コンテナ

HMI コンテナは、直感的で効率的なヒューマンマシンインターフェースを実現するためのソリューションです。このコンテナを活用することで、以下のような機能を簡単に利用できます。

### 1. HMI の作成と表示

HMI コンテナは、プログラミングを必要とせず、HMI を作成・表示することができます。誰でも簡単に高品質なインターフェースを構築可能です。

### 2. タイルの配置

コンテナ内では「タイル」と呼ばれるコンポーネントを自由に配置できます。これにより、ボタン、ランプ、チャート表示、ログ表示など多彩な機能を組み込むことが可能です。

### 3. RT-edge タグの統合

各タイルには RT-edge のタグを割り当てることができ、リアルタイムでデータの表示や書き込みを行うことが可能です。これにより、データのモニタリングや操作がスムーズに行えます。

### 4. Web ブラウザによる表示

作成した HMI は Web ブラウザを使用して表示されます。このため、専用アプリケーションを必要とせず、汎用性の高い環境で利用できます。

### 5. 外部デバイスからのアクセス

作成した HMI は PC やスマートフォンなど、外部デバイスからのアクセスにも対応しています。これにより、遠隔地からでもシームレスに操作やモニタリングを行うことができます。

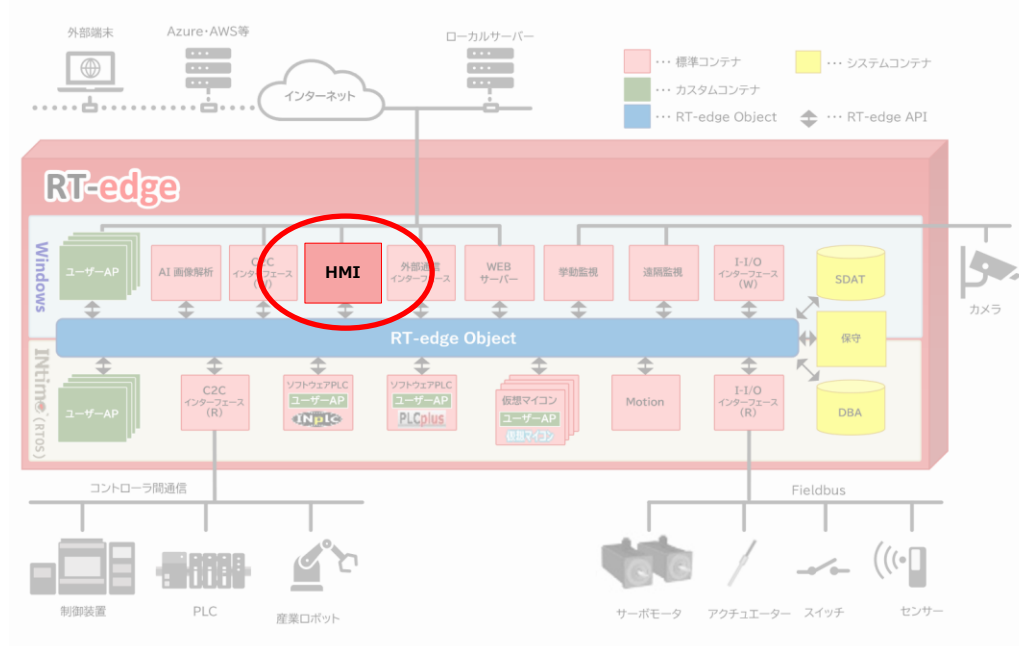


図 4. RT-edge 相関図

## 構成要素

HMI コンテナは以下コンポーネントから構成されます：

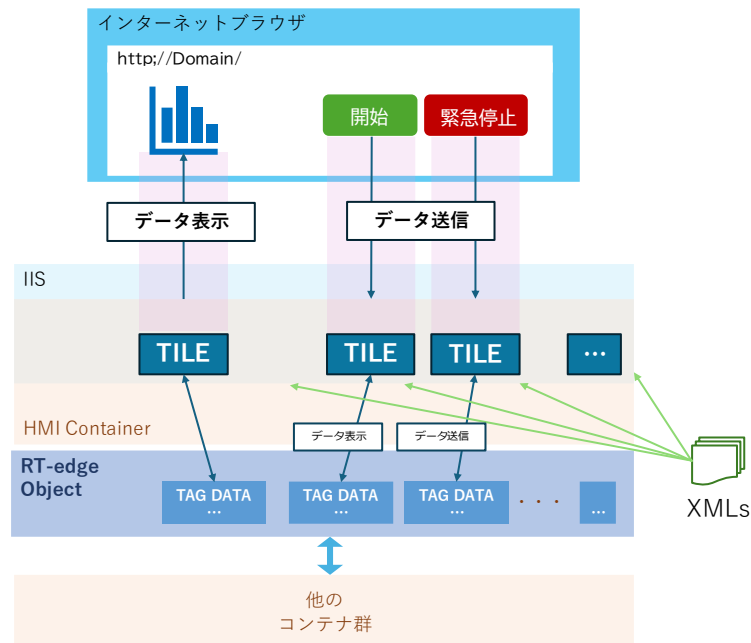


図 5. HMI コンテナ構成

表 3. HMI コンテナ構成要素

コンポーネント	内容
HMI_Server.exe	RT-edge Object のデータ入出力を行うモジュールです。
HMI_Base.dll	IIS で動作する w3wp.exe(IIS ワーカープロセス)から呼び出される dll。タイルをロードし、画面を構成するモジュールです。
HMI_{タイル名・識別名}.dll	画面上に表示するタイルモジュールです。ボタンタイルや、ランプタイル、チャートタイルなどがあります。
RT-edge コンテナ設定情報(ECI)	HMI_Server の設定を含む設定情報です。



本ドキュメントでは、主に HMI コンテナの利用方法について説明します。RT-edge 基本ソフトウェア、他サービスコンテナについては各々のマニュアルを参照ください。

## 2. 仕様

### 2.1. 動作環境

- RT-edge 基本ソフトウェアバージョン 3.6.0 以降
- Windows OS x64
- .NET 8 ランタイム (実行環境)
- .NET 8 SDK (開発環境)
- Microsoft Edge , Google Chrome (表示環境)

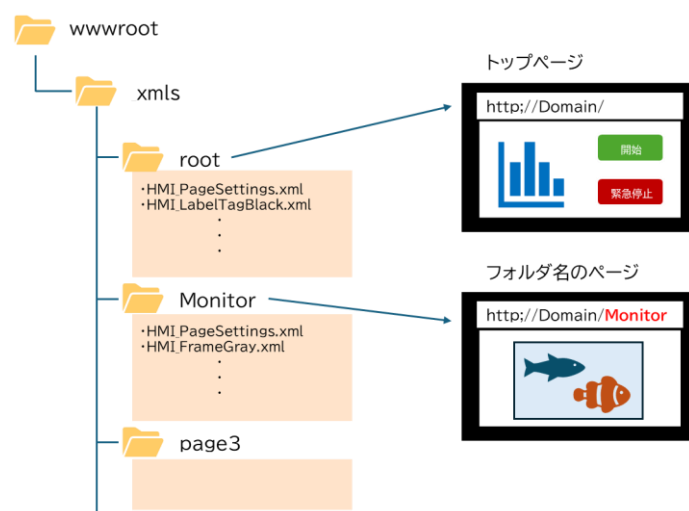
### 2.2. 使用可能タグ数

使用可能なタグ数に制限はございませんが、タグ数が多くなるとパフォーマンスに影響を及ぼす可能性があります。効率的な運用のために、適切な数のタグをご利用いただくことをおすすめいたします。

### 2.3. 画面仕様/ページ

HMI コンテナは、IIS を利用して Web ブラウザ上で表示する画面を作成するためのコンテナであり、またその画面を動作させるためのコンテナでもあります。

画面は、タイルと呼ばれるオブジェクトを配置定義し、作成します。ページ構成は、IIS で公開する `wwwroot/_xmls/{ページ名}` フォルダを作成し、各種タイル定義ファイル(xml)を配置することでページを作成することができます。



例) ページ名 : Monitor

wwwroot/\_xmls/Monitor フォルダを作成

上記フォルダ内に使用するタイル定義 xml ファイルを配置する

wwwroot/\_xmls/Monitor/HMI\_PageSettings.xml ... ページ設定用ファイル(必須)

wwwroot/\_xmls/Monitor/HMI\_LabelTagBlack.xml ... ラベルタイル定義ファイル

wwwroot/\_xmls/Monitor/HMI\_ChartLineWhite.xml ... チャートタイル定義ファイル

## 2.4. タイル

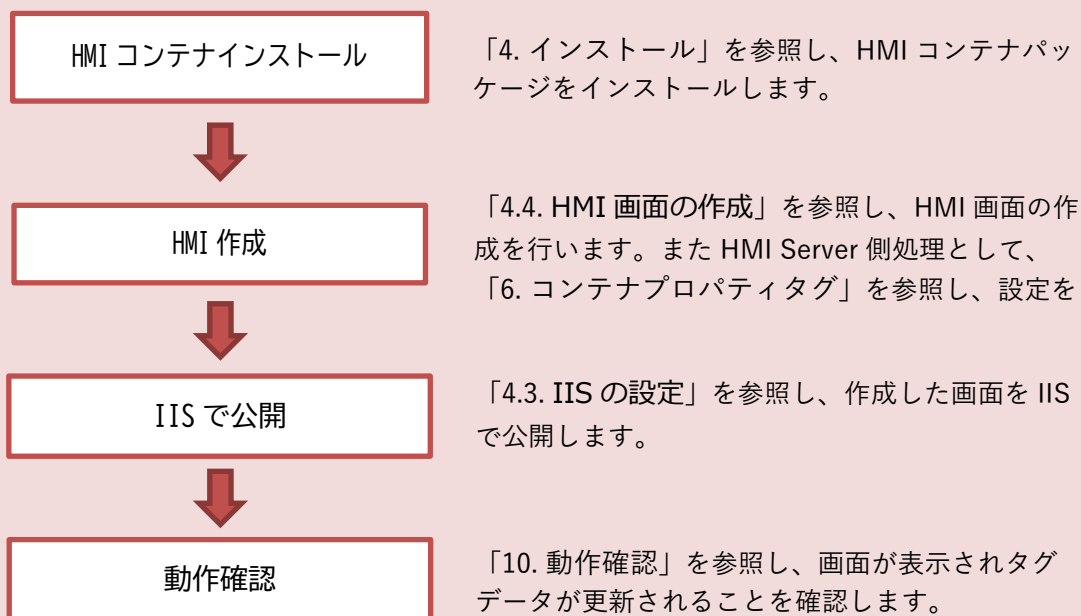
使用可能なタイルの一覧は以下の通りです。

タイル名	種類	説明
<a href="#">ラベル</a>	固定ラベル	指定した文字列を表示します。
	Tag 値参照ラベル	Tag の値を参照し表示します。
<a href="#">ボタン</a>	ON(1)設定ボタン	必ず ON(1)を設定します。
	ON(1)/OFF(0)設定ボタン	OFF 時は ON へ ON 時は OFF に設定します。 現在の状態は Tag の値を参照し表示します。
	リンクボタン	ボタン押下時、指定のページへ移行します。
<a href="#">ランプ</a>		ON(1)時は ON の色を、OFF(0)時は OFF の色を表示します。
<a href="#">チャート</a>	折れ線グラフ	タグの値を参照し折れ線グラフを表示します。
	棒グラフ	タグの値を参照し棒グラフを表示します。
<a href="#">ログ表示</a>	テーブル形式	タグの値を参照し表形式でログを表示します。
<a href="#">インナーフレーム(IFrame)</a>		タグ、または固定アドレスのページ情報・画像を表示します。
<a href="#">テキストボックス</a>	シングル	テキストボックス+設定ボタン付きのタイルです。 テキストボックスに値を入力し、設定ボタン押下でタグに値を書き込みます。

カスタムタイルを作成する場合は、「11. カスタムタイルの作成」を参照ください。

# 3. コンテナ導入フロー

本マニュアルで解説



## 4. インストール

予め RT-edge を「C:\RTedge」にインストールした状態を前提としています。

### 4.1. ファイル

HMI コンテナパッケージには以下のファイルコンポーネントが含まれています：

表 4.HMI コンテナパッケージコンポーネント（全コンポーネント）

配置先 フォルダ階層	ファイル名	説明
RTedge¥bin	HMI_Serv.exe	IIS 上で動作するプログラムと連携するプロセスです。
RTedge¥bin	HMI_Serv.xml	HMI コンテナ用 RT-edge 設定情報です。本設定ファイルを元に本コンテナの動作設定を行います。
RTedge¥bin	*.dll	HMI_Serv.exe を動作させるために必要なファイルです。
RTedge¥bin	*.xml	HMI_Serv.exe を動作させるために必要なファイルです。
RTedge¥HMI_publish	HMI_Base.dll	IIS 上で動作するプログラムです。本プログラムを使用し、Web ブラウザで画面を表示します。
RTedge¥HMI_publish	HMI_Base.exe	debug 用に使用します。
RTedge¥HMI_publish	*.dll	IIS 上で動作するプログラムの為に必要なファイルです。
RTedge¥HMI_publish	*.xml	IIS 上で動作するプログラムの為に必要なファイルです。
RTedge¥HMI_publish¥wwwroot	_content/*	同上
RTedge¥HMI_publish¥wwwroot	css/*	同上
RTedge¥HMI_publish¥wwwroot	lib/*	同上
RTedge¥HMI_publish¥wwwroot	_xmls/*	各ページを構成するファイル群です。_xmls 以下のファイルを編集することでページ内のコンテンツを変更・作成します。
RTedge¥Project¥*	*	カスタムタイトル作成用 HMI コンテナプロジェクトファイル一式

## 4.2. ファイルのインストール

HMI コンテナ用のモジュールインストール方法については、別紙インストール手順書\_HMI を参照ください。

## 4.3. IIS の設定

外部へ公開するために IIS を設定します。

1. [Windows の機能]の画面で[インターネットインフォメーションサービス]の下記図の項目にチェックをいれて機能を有効にします。

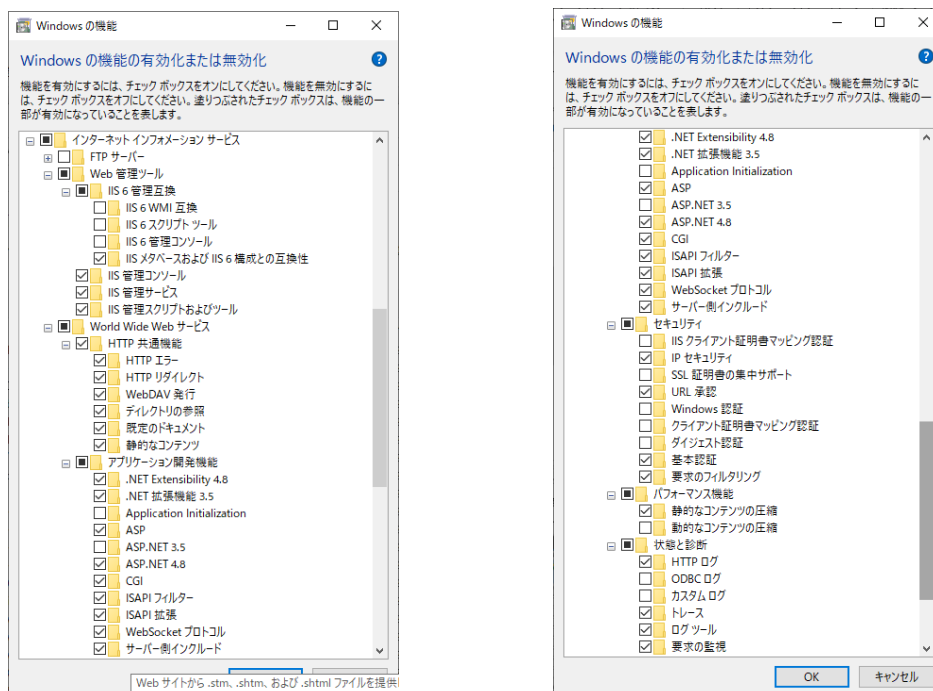


図 6 Windows の機能

2. 検索に[iis]と入力して[IIS マネージャー]を選択します。
3. [IIS マネージャー] 画面で[サイト]を選択して右クリックして、[Web サイトの追加]を選択します。

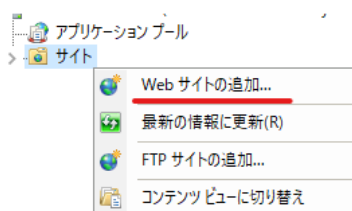




図 7 IIS の起動

4. [Web サイトの追加]画面で下記情報を入力して登録します。OK を押下することで入力完了となり、動作が開始します。

サイト名：任意のサイト名

物理パス：HMI コンテナ Base ファイル配置の HMI\_publish フォルダ

IP アドレス/ポート番号：任意の IP アドレス、ポート番号

図 8 Web サイトの追加

以上で登録は完了です。IIS を起動し、登録したサイト(IP:ポート番号)にアクセスしてください。



サイトが表示されない場合、必要なライブラリがインストールされていない可能性があります。インストール手順書に記載されている dotnet ライブラリをインストールしてください。

## 4.4. HMI 画面の作成

HMI 画面の作成は、タイル毎に参照する xml ファイルを編集します。以下にトップページを校正する手順を記載します。

### ファイル配置場所

トップページの XML ファイル配置場所は以下の通りです。

<IIS 配置場所フォルダ>%wwwroot%\\_xmls%root%

### 配置するファイルと編集

配置するファイルは、使用するタイル毎に異なります。使用可能なタイルと関連する xml は「8. タイル一覧/タイル設定値」に記載しております。

例) ラベルタイル(固定ラベル)を表示対象とする

以下にラベルタイル(固定ラベル 黒色)の xml を配置します。

<IIS 配置場所フォルダ>%wwwroot%\\_xmls%root%\HMI\_LabelFixBlack.xml

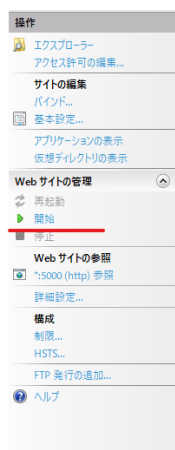
XML 内に以下記述を行うことで、2 つのラベルが画面上に配置されます。

```
<HMI_LabelFixBlack_Xml>
  <HMI_LabelFixBlacks>
    <HMI_LabelFixBlack Text="Text1" FontSize="18px" Top="70px" Left="260px" Width="150px"/>
    <HMI_LabelFixBlack Text="Text2" FontSize="18px" Top="270px" Left="260px" Width="150px"/>
  </HMI_LabelFixBlacks>
</HMI_LabelFixBlack_Xml>
```

同様に他のファイルも xml を編集することで画面へ表示できます。

### 表示確認

IIS を起動状態にし、実行モジュール、ならびに xml ファイルを IIS 配置フォルダへ配置することで動作します。また IIS マネージャー 操作パネルより、「Web サイトの管理」-「開始」で操作することも可能です。



## 4.5. 起動設定

RT-edge におけるサービスコンテナ、および関連サービス・アプリケーションの設定は、RT-edge ブートストラッパー設定により行います。HMI コンテナでは Server 部のみ RT-edge 側での起動設定を行います。これは RT-edge ブートストラッパー設定に準拠します。Base 部は前章で説明した通りです：

- 1) C:\RT-edge\bin\EgBoot.xml をテキストエディタで開きます。
- 2) RTedge エlement内の Services Element内に、HMI コンテナ用のElement(Service Element)を追加します。Service Name にはサービスコンテナ名としてユニークになる任意の名前を指定します。

また Argument 属性には、HMI コンテナ起動引数が設定可能です。

※通常は未指定で問題ありません。必要に応じて定義してください。

引数	値	説明
-Debug	なし	デバッグコード表示用コンソール画面を表示します。

- 3) 以下では、HMI コンテナの起動設定の内容です。

```
<?xml version="1.0" encoding="utf-8"?>
<RTedge xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  :
  <Services>
    <Service Name ="EgBoot" Argument="RTCD=NodeA;TagMaxNum=10000" >
    </Service>
    <Service Name="EgLog" Path="EgLog.exe" Argument="DispNumMax=500" >
    </Service>
    <Service Name="EgTime" Path="EgTime.exe" >
    </Service>
    <Service Name="HMI_Server" Path="HMI_Server.exe" ECI="HMI_Server.xml" >
    </Service>
  </Services>
</RTedge>
```

図 9.HMI コンテナ起動登録

## 4.6. 動作確認

RT-edge ソフトウェアを起動し、HMI コンテナ(サーバー部、Base 部)が正常に起動することを確認します:

<HMI コンテナ サーバー部>

- 1) RT-Edge ソフトウェア(C:\¥RTedge¥bin¥EgBoot.exe)を管理者権限で開始します。
- 2) RT-Edge オブジェクトブラウザ(C:\¥RTedge¥bin¥EgBrow.exe)を起動します。



RT-edge 起動直後に RT-edge オブジェクトブラウザを起動すると初期化中の為、想定されるタグが表示されない場合があります。一度 RT-edge オブジェクトブラウザを終了し、再度起動させてください。

- 3) サービスインジケータタグから、正常状態であることを確認します(サービスインジケータタグについては「10.2. サービスインジケータタグ」を参照ください)。

### 正常状態

- SERVICE.HMI\_Server.Status が **01(1)** であること
- SERVICE.HMI\_Server.Run が **True(1)** であること
- SERVICE.HMI\_Server.Error が **False(0)** であること
- SERVICE.HMI\_Server.Live が **増加**していくこと

RT-edge Object Browser

ファイル

Tags Containers Collectors Datasets Mailboxes

Name	Current Value	Type	Source	Comment
SERVICE.HMI_Serv.Error	False (0)	bool		
SERVICE.HMI_Serv.Live	000038e1 (14561)	UInt32		
SERVICE.HMI_Serv.Run	True (1)	bool		
SERVICE.HMI_Serv.Status	01 (1)	byte		

図 10. RT-edge オブジェクトブラウザ起動時の様子

上記の状態になっていない場合には、以下のトラブルシューティングをご参照ください。

「サービスインジケータの.Error が true になっています。」

「サービスインジケータの.Run が false になっています。」



SERVICE.HMI\_Server.Run タグは、SERVICE.HMI\_Server.AutoRun タグの値が True または、メッセージ通信において EM\_SERVICE\_RUN メッセージ(102)送信後に True になります。

<HMI コンテナ Base 部>

1) IIS で Web サイトを開始状態にします。

2) ブラウザ上で公開ページの URL を指定し表示します。

IIS 上から「Web サイトの参照」 - 「\*:IP(http)参照」から  
も表示ができます。



3) ブラウザ上にタイルが表示され、タグのデータの表示、書き込みが正常に行えるか確認します。

値の変更・確認は RT-Edge オブジェクトブラウザ(C:\RTedge\bin\EgBrow.exe)を起動し  
て行ってください。



- ・ 起動が出来ない場合はここからインストーラーをダウンロードの上、インストールしてください。  
<https://dotnet.microsoft.com/permalink/dotnetcore-current-windows-runtime-bundle-installer>
- ・ タグのデータを表示するには [IIS 配置フォルダ]/appsettings.json の UpdateTile を True にする必要があります。

## 4.7. 終了設定

RT-edge におけるサービスコンテナ、および関連サービス・アプリケーションの終了は、RT-edge 終了サービス「EgShDown」により行います。HMI コンテナ サーバー部の終了設定も同様、RT-edge 終了サービス設定に準拠します。Base 部は、IIS 上で停止を行います。以下はサーバー部の説明です：

- 1) C:\RTedge\bin\EgShDown.xml をテキストエディタで開きます。
- 2) ArrayOfAnyType エlement 内に、EgBoot.xml に記載した挙動監視コンテナの Service Name を追加します。

```
<anyType xsi:type="xsd:string">HMI_Server</anyType>
```

- 3) 編集を保存し、ファイルを閉じます。
- 4) 追加結果は以下のようになります。

```
...:
<ArrayOfAnyType ...>
  <anyType xsi:type="xsd:string">HMI_Server</anyType>
  <anyType xsi:type="xsd:string">EgLog</anyType>
  <anyType xsi:type="xsd:string">EgTime</anyType>
  <anyType xsi:type="xsd:string">EgBoot</anyType>
</ArrayOfAnyType>
```

図 11.HMI コンテナ終了登録

RT-edge ソフトウェアの終了時は、C:\RTedge\bin\EgShDown.exe を実行します。

## 5. 設定概要

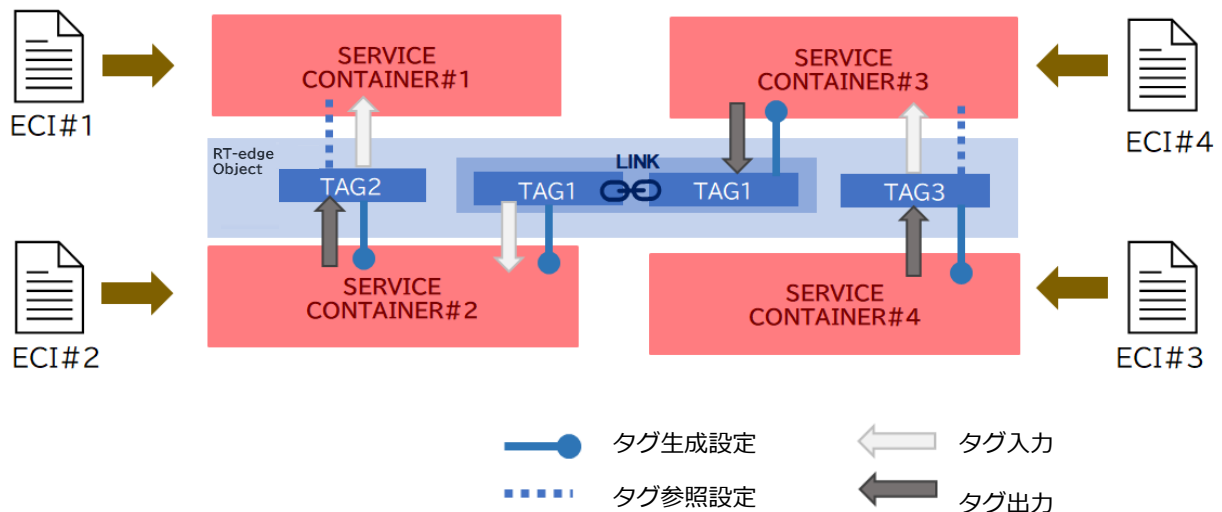
サービスコンテナ設定により担当するターゲットのデータとタグとの接続が可能となります。

サービスコンテナにおける基本設定は、タグ、データセットの定義を主とした RT-edge Object 設定と、サービスコンテナの入出力周期やプライオリティ設定等、コンテナプロパティ設定に分類されます。

設定項目	説明
RT-edge Object 設定	<p><u>タグ設定</u></p> <p>ローカルタグ生成設定・リンクタグ生成設定</p> <p>タグ参照設定</p> <p><u>データセット設定</u></p> <p>タグ・コレクション定義</p> <p>周期・プライオリティ設定</p>
コンテナプロパティ設定	<p>データ更新方式(オンデマンド・サイクリック(周期設定))</p> <p>プライオリティ設定等</p> <p>※コンテナプロパティ値は各サービスコンテナにより実装が異なります。</p>

### 5.1. ECI 設定

RT-edge Object 設定、プロパティ設定は、コンテナ毎に定義する設定情報(ECI: RT-edge コンテナ設定情報)に基づきます。ECI ファイルは XML 形式のテキストファイルとして生成されています。



各サービスコンテナは、タグ・リンクタグ生成設定を行います。

タグに対する入出力方向設定・参照設定を行います。

### 5.1.1. RT-edge Object 設定

RT-edge Object 設定は、ECI ファイルを編集します。XML タグの編集要素は以下のように定義されています:

設定項目	設定手順												
タグ設定	<p><u>ローカルタグ</u></p> <p>生成 &lt;Tags&gt;エレメント内に、&lt;Tag&gt;を生成します。 サービスコンテナ独自の名称(一意名)で&lt;Tag Name=&gt;の設定を行います。</p> <p><u>リンクタグ</u></p> <p>&lt;Tags&gt;エレメント内に、&lt;Tag&gt;を生成します。 他サービスコンテナの提供するタグと同名で&lt;Tag Name=&gt;の設定を行います。</p> <p><b>Tag</b></p> <table> <tr> <th>キーワード</th><th>説明</th></tr> <tr> <td>Name</td><td>公開するタグ名を設定します。</td></tr> <tr> <td>Type</td><td>RT-edge データ型に関連する型定義値を設定します。 ※参照: データ型</td></tr> <tr> <td>Size</td><td>タグデータサイズを指定します。 ※参照: データ型</td></tr> <tr> <td>Address</td><td>サービスコンテナにおけるデータ取得元、宛先となるアドレス情報を指定します。本アドレス書式は、サービスコンテナ毎に異なります。</td></tr> <tr> <td>Comment</td><td>タグに対するコメントを設定します。</td></tr> </table>	キーワード	説明	Name	公開するタグ名を設定します。	Type	RT-edge データ型に関連する型定義値を設定します。 ※参照: データ型	Size	タグデータサイズを指定します。 ※参照: データ型	Address	サービスコンテナにおけるデータ取得元、宛先となるアドレス情報を指定します。本アドレス書式は、サービスコンテナ毎に異なります。	Comment	タグに対するコメントを設定します。
キーワード	説明												
Name	公開するタグ名を設定します。												
Type	RT-edge データ型に関連する型定義値を設定します。 ※参照: データ型												
Size	タグデータサイズを指定します。 ※参照: データ型												
Address	サービスコンテナにおけるデータ取得元、宛先となるアドレス情報を指定します。本アドレス書式は、サービスコンテナ毎に異なります。												
Comment	タグに対するコメントを設定します。												
参照	<p><u>入力参照</u>: &lt;TagRefs_IN&gt; エレメント内</p> <p><u>出力参照</u>: &lt;TagRefs_OUT&gt; エレメント内</p> <p>&lt;TagRef Name=&gt;に参照するタグを指定します。</p> <p><b>TagRef</b></p> <table> <tr> <th>キーワード</th><th>説明</th></tr> <tr> <td>Name</td><td>参照するタグを指定します。</td></tr> </table>	キーワード	説明	Name	参照するタグを指定します。								
キーワード	説明												
Name	参照するタグを指定します。												
データセット設定	<p>&lt;Datasets&gt;エレメント内に、&lt;Dataset&gt;</p> <p>タグ・コレクション定義 &lt;Dataset Name=&gt;にデータセット名を指定します。</p> <p>Dataset エレメント内に、&lt;TagRefs&gt;エレメントを作成します。</p> <p>&lt;TagRef Name=&gt;に参照するタグを指定します。</p> <p><b>TagRef</b></p> <table> <tr> <th>キーワード</th><th>説明</th></tr> <tr> <td>Name</td><td>参照するタグを指定します。</td></tr> </table> <p>※Dataset 内に TagRef オブジェクトを列挙します。</p>	キーワード	説明	Name	参照するタグを指定します。								
キーワード	説明												
Name	参照するタグを指定します。												



設定項目	設定手順										
周期・プライオリティ 設定	<p>&lt;Collectors&gt;エレメント内に、&lt;Collector&gt;を作成し</p> <p>&lt;Collector Name=&gt;に名称を設定します (Dataset を収集する機能名)</p> <p><b>Collector</b></p> <table> <tr> <th>キーワード</th><th>説明</th></tr> <tr> <td>Name</td><td>Dataset 収集機能名を指定します</td></tr> <tr> <td>Interval</td><td>収集周期を指定します (1ms 単位)</td></tr> <tr> <td>Priority</td><td>プライオリティを設定します</td></tr> <tr> <td>DatasetName</td><td>収集するデータセット名を指定します。</td></tr> </table>	キーワード	説明	Name	Dataset 収集機能名を指定します	Interval	収集周期を指定します (1ms 単位)	Priority	プライオリティを設定します	DatasetName	収集するデータセット名を指定します。
キーワード	説明										
Name	Dataset 収集機能名を指定します										
Interval	収集周期を指定します (1ms 単位)										
Priority	プライオリティを設定します										
DatasetName	収集するデータセット名を指定します。										

### 5.1.2. コンテナプロパティ設定

サービスコンテナプロパティ設定値は、Tag として登録されており、サービスコンテナ実装毎に数や種類は異なります。規定値プロパティは、SERVICE.キーワードをプリフィックスとしたタグ名で登録されています:

設定項目	設定手順												
コンテナプロパティ値	<p>&lt;Tags&gt;エレメント内に、&lt;Tag&gt;において、SERVICE.キーワードをプリフィックスとしたタグは、コンテナプロパティタグです:</p> <p><b>Tag</b></p> <table> <tr> <th>キーワード</th><th>説明</th></tr> <tr> <td>Name</td><td>SERVICE. キーワードをプリフィックスとした名称で設定されています。</td></tr> <tr> <td>Type</td><td>RT-edge データ型に関連する型定義値を設定します。</td></tr> <tr> <td>Size</td><td>タグデータサイズを指定します。</td></tr> <tr> <td>Value</td><td>設定値</td></tr> <tr> <td>Comment</td><td>タグに対するコメントを設定します。</td></tr> </table>	キーワード	説明	Name	SERVICE. キーワードをプリフィックスとした名称で設定されています。	Type	RT-edge データ型に関連する型定義値を設定します。	Size	タグデータサイズを指定します。	Value	設定値	Comment	タグに対するコメントを設定します。
キーワード	説明												
Name	SERVICE. キーワードをプリフィックスとした名称で設定されています。												
Type	RT-edge データ型に関連する型定義値を設定します。												
Size	タグデータサイズを指定します。												
Value	設定値												
Comment	タグに対するコメントを設定します。												



コンテナプロパティ設定については、各サービスコンテナに付与するユーザーズマニュアルを参照してください。

## 6. コンテナプロパティタグ

HMI コンテナにおける RT-edge Object 設定は行いません。コンテナプロパティ設定では起動設定のみ記載します。

### 6.1. 一覧

HMI コンテナには、以下のプロパティ設定があります。サービスコンテナのプロパティ設定はタグの一部として構成されています。以下にコンテナプロパティ定義を記載します。

表 5. サービスプロパティタグ一覧

プロパティ Tag 名	概要
SERVICE.<サービス名>.AutoRun	サービスコンテナ起動時に IIS とのデータやり取りを有効にします。



コンテナプロパティタグはあらかじめ定義されており、名称は、SERVICE. から開始しています。

### 6.2. サービス動作自動スタート指定(.AutoRun)

Tag		備考
Name	SERVICE.HMI_Server.AutoRun	
Type	Bool(1)	
Size	1	
Value	1	初期値
値	意味	備考
0 (false)	サービス起動時に IIS とのデータやり取りを有効にしません。	
1 (true)	サービス起動時に IIS とのデータやり取りを有効にします。	

## 7. HMI アプリケーション設定/ページ設定

HMI コンテナ実行環境におけるアプリケーション設定や表示するページ設定に関して記載します。

### 7.1. アプリケーション設定

アプリケーション全体の設定情報は以下ファイルに記載があります。以下ファイルを編集することで設定の変更が可能です。設定変更はアプリケーションの再起動が必要です。

[IIS 配置フォルダ]/appsettings.json

```
{
  "AppSettings": {
    "OutputLogFile": true,
    "OutputLogMode": 3,
    "DisplayCoord": false,
    "UpdateTile": false
  },
  "AuthenticationSettings": {
    "IsLoginEnabled": false,
    "AllowedUserName": "test",
    "AllowedPassword": "test"
  }
}
```

#### 7.1.1. ログ出力レベル指定(AppSettings – OutputLogFile)

項目名		
OutputLogFile		ログをファイルへ出力するか否か設定
値	意味	備考
true	ログをファイルへ出力する	
false	ログをファイルへ出力しない	

#### 7.1.2. ログ出力レベル指定(AppSettings – OutputLogMode)

項目名		
OutputLogMode		ログ出力レベルを指定
値	意味	備考
1	ERROR 情報出力	ビット演算で出力
2	DEBUG 情報出力	ビット演算で出力

### 7.1.3. X/Y 座標表示(AppSettings – DisplayCoord)

項目名		
DisplayCoord	ブラウザ上でマウスカーソルに追従して X, Y 座標を表示するか否かの設定 (HMI 作成時用)	
値	意味	備考
true	マウスカーソルに追従して X/Y 座標を表示する	表示時はボタンなど押下できません
false	マウスカーソルに追従して X/Y 座標を表示しない	

### 7.1.4. タグデータ表示(AppSettings – UpdateTile)

項目名		
UpdateTile	Edge にアクセスして Tile の情報を更新するか否か設定	
値	意味	備考
true	タグのデータを参照する	サーバーを起動していない場合は起動が遅くなります
false	タグのデータを参照しない	

### 7.1.5. ログイン画面 使用有無フラグ(AuthenticationSettings – IsLoginEnabled)

項目名		
IsLoginEnabled	ログイン画面を使用するか否かを設定	
値	意味	備考
true	初回表示時ログイン画面を表示する	
false	ログイン画面は表示しない	

### 7.1.6. ログインユーザ名(AuthenticationSettings– AllowedUserName)

項目名		
AllowedUserName	ログイン画面使用時のログインユーザ名を設定(現状 1 ユーザのみ)	
値	意味	備考
任意の値		

### 7.1.7. ログインパスワード(AuthenticationSettings– AllowedPassword)

項目名		
-----	--	--

AllowedPassword			ログイン画面使用時のログインパスワードを設定(現状 1 ユーザのみ 空無し)
値	意味	備考	
任意の値			

## 7.2. ページ設定

ページ毎の設定情報は以下ファイルに記載します。以下ファイルを編集することで設定の変更が可能です。

[IIS 配置フォルダ]/wwwroot/\_xmls/root(またはページ名)/HMI\_PageSettings.xml

```
<PageSettings>
  <PageTitle>root</PageTitle>
  <BackgroundColor>white</BackgroundColor>
  <BackgroundImage>ikesu_back.png</BackgroundImage>
  <BackgroundSize>cover</BackgroundSize>
  <PageHeight_Px>1500px</PageHeight_Px>
</PageSettings>
```

### 7.2.1. ページタイトル (PageTitle)

項目名		
PageTitle	ページタイトルを指定	
値	意味	備考
文字列	ページタイトルを指定	

### 7.2.2. 背景色 (BackgroundColor)

項目名		
BackgroundColor	ページ背景色を指定	
値	意味	備考
<ul style="list-style-type: none"> <li>・名前付きの色 (Blue 等)</li> <li>・RGB 16 進値 (#f09 等)</li> <li>・その他 CSS で指定可能な値</li> </ul>	背景色を指定	CSS で指定可能なフォーマットで記載 色の表示範囲は PageHeight_Px で指定する 必要あり (記載が無い場合、色は表示されません)

### 7.2.3. 背景画像 (BackgroundImage)

項目名		
BackgroundImage	ページ背景画像を指定	
値	意味	備考
画像ファイルパス名	画像ファイルパス名を	wwwroot/以下のファイルパス名を相対パスで指定 画像のサイズは BackgroundSize で指定、 表示範囲は PageHeight_Px で指定する必要あり (記載が無い場合、画像は表示されません)

#### 7.2.4.背景画像のサイズ (BackgroundSize)

項目名		
BackgroundSize		ページ背景画像のサイズプロパティを指定
値	意味	備考
cover	背景画像を要素のサイズに完全に覆うように拡大縮小	
contain	背景画像を要素内に完全に収めるように縮小または拡大	
その他 CSS で background-size に 指定可能な設定値	CSS で指定可能な設定値を 指定 例) 画像の元のサイズ auto	表示するブラウザによって適用されない場合があります。 また本設定に関わらず表示範囲は PageHeight_Px で指定する必要あり

#### 7.2.5.背景表示範囲 (PageHeight\_Px)

項目名		
PageHeight_Px		ページ背景表示範囲(高さ)を指定
値	意味	備考
1200px 等	背景色・画像を表示する高さを px 単位で指定	

## 8. タイル一覧/タイル設定値

HMI コンテナで使用するタイルとその設定値に関して記載します:

### 8.1. ラベル

ラベルのタイルを記載します:

#### 8.1.1. ラベル 固定ラベル

固定ラベル

タイル名	種類	備考	説明
ラベル	固定ラベル		指定した文字列を表示
対象 XML	黒色	HMI_LabelFixBlack.xml	
イメージ	XML 要素名/属性名		説明
固定ラベル	HMI_LabelFixBlack		タイルを定義する要素名
	Text		表示する固定ラベル
	Top		画面上からの位置(px 指定)
	Left		画面左からの位置(px 指定)
	FontSize		ラベルサイズ
	Width		表示幅
	Zindex		要素の前後関係(値が大きいほど手前に、小さいほど奥に表示)

Xml サンプル

```
<HMI_LabelFixBlack_Xml">
  <HMI_LabelFixBlacks>
    <HMI_LabelFixBlack Text="固定ラベル" FontSize="40px" Top="50px" Left="50px" Width="500px"/>
  </HMI_LabelFixBlacks>
</HMI_LabelFixBlack_Xml>
```

D設備

対象 XML	赤色	HMI_ButtonCircleGreen.xml	
イメージ		意味	備考
		HMI_LabelFixRed	タイルを定義する要素名
		Text	表示する固定ラベル
		Top	画面上からの位置(px 指定)
		Left	画面左からの位置(px 指定)
		FontSize	ラベルサイズ
		Width	表示幅
		Zindex	要素の前後関係(値が大きいほど手前に、小さいほど奥に表示)

Xml サンプル

```
<HMI_LabelFixRed_Xml>
  <HMI_LabelFixReds>
    <HMI_LabelFixRed Text="D 設備" FontSize="30px" Top="110px" Left="500px" Width="100px" />
  </HMI_LabelFixReds>
</HMI_LabelFixRed_Xml>
```



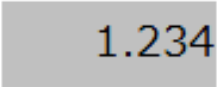
### 8.1.2. ラベル タグラベル

タイル名	種類	備考	説明
ラベル	タグラベル	指定したタグの値を表示	

対象 XML	黒色	HMI_LabelTagBlack.xml
--------	----	-----------------------

イメージ	XML 要素名/属性名	説明
	HMI_LabelTagBlack	タイルを定義する要素名
	Tag	表示に使用するタグ名
	Top	画面上からの位置(px 指定)
	Left	画面左からの位置(px 指定)
	FontSize	ラベルサイズ
	Height	表示高さ
	Width	表示幅
	Zindex	要素の前後関係(値が大きいほど手前に、小さいほど奥に表示)

[Xml サンプル]

```
<HMI_LabelTagBlack_Xml>
  <HMI_LabelTagBlacks>
    <HMI_LabelTagBlack Tag="Temp_01" FontSize="20px" Top="120px" Left="900px" Width="100px" Height="40px" />
  </HMI_LabelTagBlacks>
</HMI_LabelTagBlack_Xml>
```

[備考] 背景グレーは固定

対象 XML	赤色	HMI_LabelTagRed.xml
--------	----	---------------------

イメージ	意味	備考
	HMI_LabelTagRed	タイルを定義する要素名
	Tag	表示に使用するタグ名
	Top	画面上からの位置(px 指定)
	Left	画面左からの位置(px 指定)
	Height	表示高さ
	FontSize	ラベルサイズ
	Width	表示幅
	Zindex	要素の前後関係(値が大きいほど手前に、小さいほど奥に表示)

Xml サンプル

```
<HMI_LabelTagRed_Xml>
  <HMI_LabelTagReds>
    <HMI_LabelTagRed Tag="Temp_01" FontSize="20px" Top="120px" Left="1100px" Width="100px" Height="40px" />
  </HMI_LabelTagReds>
</HMI_LabelTagRed_Xml>
```

[備考] 背景グレーは固定

## 8.2. ボタン

ボタン形状・動作を行うタイルです:

### 8.2.1. ボタン ON 設定 | 円形

タイル名	種類	備考	説明
ボタン	ON 設定	形状: 円	ON(1)を設定 (OFF 設定は別ボタン)

対象 XML	緑	HMI_ButtonCircleGreen.xml
イメージ	XML 要素名/属性名	説明



HMI_ButtonCircleGreen	タイルを定義する要素名
Tag	ON の設定に使用する Tag
Text	表示する固定ラベル
Top	画面上からの位置(px 指定)
Left	画面左からの位置(px 指定)
FontSize	ラベルサイズ
Width	表示幅
Zindex	要素の前後関係(値が大きいほど手前に、小さいほど奥に表示)

Xml サンプル

```
<HMI_ButtonCircleGreen_Xml>
  <HMI_ButtonCircleGreens>
    <HMI_ButtonCircleGreen Tag="Emergency_01" Text="停止 1" FontSize="15px" Top="310px" Left="500px" Width="55px" />
  </HMI_ButtonCircleGreens>
</HMI_ButtonCircleGreen_Xml>
```


対象 XML	赤色	HMI_ButtonCircleRed.xml
イメージ	意味	備考



HMI_ButtonCircleRed	タイルを定義する要素名
Tag	ON の設定に使用する Tag
Text	表示する固定ラベル
Top	画面上からの位置(px 指定)
Left	画面左からの位置(px 指定)
FontSize	ラベルサイズ
Width	表示幅
Zindex	要素の前後関係(値が大きいほど手前に、小さいほど奥に表示)


Xml サンプル

```
<HMI_ButtonCircleRed_Xml >
  <HMI_ButtonCircleReds>
    <HMI_ButtonCircleRed Tag="Emergency_02" Text="停止 2" FontSize="15px" Top="375px" Left="500px" Width="55px" />
  </HMI_ButtonCircleReds>
</HMI_ButtonCircleRed_Xml>
```

対象 XML	濃青	HMI_ButtonCircleRoyalblue.xml	
イメージ	XML 要素名/属性名		説明
	HMI_ButtonCircleRoyalblue		タイルを定義する要素名
	Tag		ON の設定に使用する Tag
	Text		表示する固定ラベル
	Top		画面上からの位置(px 指定)
	Left		画面左からの位置(px 指定)
	FontSize		ラベルサイズ
	Width		表示幅
	Zindex		要素の前後関係(値が大きいかほど手前に、小さいほど奥に表示)


#### Xml サンプル

```
<HMI_ButtonCircleRoyalblue_Xml>
  <HMI_ButtonCircleRoyalblues>
    <HMI_ButtonCircleRoyalblue Tag="Emergency_03" Text="停止 3" FontSize="15px" Top="440px" Left="500px" Width="55px" />
  </HMI_ButtonCircleRoyalblues>
</HMI_ButtonCircleRoyalblue_Xml>
```

対象 XML	オレンジ色	HMI_ButtonCircleOrange.xml	
イメージ	意味	備考	
	HMI_ButtonCircleOrange	タイルを定義する要素名	
	Tag	ON の設定に使用する Tag	
	Text	表示する固定ラベル	
	Top	画面上からの位置(px 指定)	
	Left	画面左からの位置(px 指定)	
	FontSize	ラベルサイズ	
	Width	表示幅	
	Zindex	要素の前後関係(値が大きいかほど手前に、小さいほど奥に表示)	


#### Xml サンプル

```
<HMI_ButtonCircleOrange_Xml>
  <HMI_ButtonCircleOranges>
    <HMI_ButtonCircleOrange Tag="Emergency_04" Text="停止 4" FontSize="15px" Top="505px" Left="500px" Width="55px" />
  </HMI_ButtonCircleOranges>
</HMI_ButtonCircleOrange_Xml>
```

対象 XML	濃紺	HMI_ButtonCircleDarkblueOver.xml	
イメージ	意味	備考	
	HMI_ButtonCircleDarkblueOver	タイルを定義する要素名	
	Tag	ON の設定に使用する Tag	
	Text	表示する固定ラベル	
	Top	画面上からの位置(px 指定)	
	Left	画面左からの位置(px 指定)	
	FontSize	ラベルサイズ	
	Width	表示幅	
	Zindex	要素の前後関係(値が大きいかほど手前に、小さいほど奥に表示)	
Xml サンプル			
<pre>&lt;HMI_ButtonCircleDarkblueOver_Xml&gt;   &lt;HMI_ButtonCircleDarkblueOvers&gt;     &lt;HMI_ButtonCircleDarkblueOver Tag="Emergency_04" Text="停止 4" FontSize="15px" Top="570px" Left="500px" Width="55px" /&gt;   &lt;/HMI_ButtonCircleDarkblueOvers&gt; &lt;/HMI_ButtonCircleDarkblueOver_Xml&gt;</pre>			
備考			
マウスオーバー時、色を変更します。			


## 8.2.2.ボタン ON 設定 | 矩形

タイル名	種類	備考	説明
ボタン	ON 設定	形状: 矩形	ON(1)を設定 (OFF 設定は別ボタン)

対象 XML	緑	HMI_ButtonRectGreen.xml	
イメージ		XML 要素名/属性名	説明
		HMI_ButtonRectGreen	タイルを定義する要素名
		Tag	ON の設定に使用する Tag
		Text	表示する固定ラベル
		Top	画面上からの位置(px 指定)
		Left	画面左からの位置(px 指定)
		Height	表示高さ
		FontSize	ラベルサイズ
		Width	表示幅
		Zindex	要素の前後関係(値が大きいかほど手前に、小さいほど奥に表示)

### Xml サンプル

```
<HMI_ButtonLampTglRectGreen_Xml >
  <HMI_ButtonLampTglRectGreens>
    <HMI_ButtonLampTglRectGreen Tag="Cmd_Stop" Text="停止" FontSize="22px" Top="310px" Left="1300px" Width="110px" Height="40px" />
  </HMI_ButtonLampTglRectGreens>
</HMI_ButtonLampTglRectGreen_Xml>
```

対象 XML	赤色	HMI_ButtonRectRed.xml	
イメージ		意味	備考
		HMI_ButtonRectRed	タイルを定義する要素名
		Tag	ON の設定に使用する Tag
		Text	表示する固定ラベル
		Top	画面上からの位置(px 指定)
		Left	画面左からの位置(px 指定)
		Height	表示高さ
		FontSize	ラベルサイズ
		Width	表示幅
		Zindex	要素の前後関係(値が大きいかほど手前に、小さいほど奥に表示)


### Xml サンプル

```
<HMI_ButtonRectRed_Xml >
  <HMI_ButtonRectReds>
    <HMI_ButtonRectRed Tag="Emergency_02" Text="停止 2" FontSize="15px" Top="375px" Left="500px" Width="55px" />
  </HMI_ButtonRectReds>
</HMI_ButtonRectRed_Xml>
```

対象 XML	濃青	HMI_ButtonRectRoyalblue.xml	
イメージ	XML 要素名/属性名		説明
	HMI_ButtonRectRoyalblue		タイルを定義する要素名
	Tag		ON の設定に使用する Tag
	Text		表示する固定ラベル
	Top		画面上からの位置(px 指定)
	Left		画面左からの位置(px 指定)
	Height		表示高さ
	FontSize		ラベルサイズ
	Width		表示幅
	Zindex		要素の前後関係(値が大きいほど手前に、小さいほど奥に表示)

#### Xml サンプル

```
<HMI_ButtonRectRoyalblue_Xml>
  <HMI_ButtonRectRoyalblues>
    <HMI_ButtonRectRoyalblue Tag="Emergency_03" Text="停止 3" FontSize="15px" Top="440px" Left="500px" Width="55px" />
  </HMI_ButtonRectRoyalblues>
</HMI_ButtonRectRoyalblue_Xml>
```

対象 XML	オレンジ色	HMI_ButtonRectOrange.xml	
イメージ	意味	備考	
	HMI_ButtonRectOrange	タイルを定義する要素名	
	Tag	ON の設定に使用する Tag	
	Text	表示する固定ラベル	
	Top	画面上からの位置(px 指定)	
	Left	画面左からの位置(px 指定)	
	Height	表示高さ	
	FontSize	ラベルサイズ	
	Width	表示幅	
	Zindex	要素の前後関係(値が大きいほど手前に、小さいほど奥に表示)	

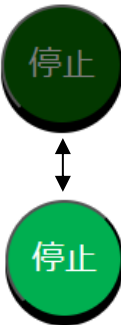
#### Xml サンプル

```
<HMI_ButtonRectOrange_Xml>
  <HMI_ButtonRectOranges>
    <HMI_ButtonRectOrange Tag="Emergency_04" Text="停止 4" FontSize="15px" Top="505px" Left="500px" Width="55px" />
  </HMI_ButtonRectOranges>
</HMI_ButtonRectOrange_Xml>
```

対象 XML	濃紺	HMI_ButtonRectDarkblueOver.xml	
イメージ		意味	備考
		HMI_ButtonRectDarkblueOver	タイルを定義する要素名
		Tag	ON の設定に使用する Tag
		Text	表示する固定ラベル
		Top	画面上からの位置(px 指定)
		Left	画面左からの位置(px 指定)
		Height	表示高さ
		FontSize	ラベルサイズ
		Width	表示幅
		Zindex	要素の前後関係(値が大きいかほど手前に、小さいほど奥に表示)
<p>Xml サンプル</p> <pre> &lt;HMI_ButtonRectDarkblueOver_Xml&gt;   &lt;HMI_ButtonRectDarkblueOvers&gt;     &lt;HMI_ButtonRectDarkblueOver Tag="Emergency_04" Text="停止 4" FontSize="15px" Top="570px" Left="500px" Width="55px" /&gt;   &lt;/HMI_ButtonRectDarkblueOvers&gt; &lt;/HMI_ButtonRectDarkblueOver_Xml&gt; </pre>			
<p>備考</p> <p>マウスオーバー時、色を変更します。</p>			




### 8.2.3.ボタン ON/OFF 設定 | 円形

タイル名	種類	備考	説明
ボタン	ON/OFF 設定	形状: 円	Tag の値が OFF(0)の場合 Off の表示 ボタン押下で ON(1)を設定 Tag の値が ON(1)の場合 On の表示 ボタン押下で OFF(0)を設定

対象 XML	緑	HMI_ButtonLampTglCircleGreen.xml	
イメージ	XML 要素名/属性名		説明
OFF(0)		HMI_ButtonLampTglCircleGreen	タイルを定義する要素名
		Tag	ON の設定に使用する Tag
		Text	表示する固定ラベル
		Top	画面上からの位置(px 指定)
		Left	画面左からの位置(px 指定)
		FontSize	ラベルサイズ
		Width	表示幅
		Zindex	要素の前後関係(値が大きいかほど手前に、小さいほど奥に表示)
ON(1)			

#### Xml サンプル



```
<HMI_ButtonLampTglCircleGreen_Xml>
  <HMI_ButtonLampTglCircleGreens>
    <HMI_ButtonLampTglCircleGreen Tag="Command_Stop_2" Text="停止" FontSize="22px" Top="600px" Left="800px" Width="80px"/>
  </HMI_ButtonLampTglCircleGreens>
</HMI_ButtonLampTglCircleGreen_Xml>
```

対象 XML	赤色	HMI_ButtonLampTglCircleRed.xml	
イメージ		意味	備考
OFF(0)		HMI_ButtonLampTglCircleRed	タイルを定義する要素名
		Tag	ON の設定に使用する Tag
		Text	表示する固定ラベル
		Top	画面上からの位置(px 指定)
ON(1)		Left	画面左からの位置(px 指定)
		FontSize	ラベルサイズ
		Width	表示幅
		Zindex	要素の前後関係(値が大きいかほど手前に、小さいほど奥に表示)

#### Xml サンプル



```
<HMI_ButtonLampTglCircleRed_Xml>
  <HMI_ButtonLampTglCircleReds>
    <HMI_ButtonLampTglCircleRed Tag="Emergency_2" Text="非常停止" FontSize="16px" Top="600px" Left="900px" Width="80px"/>
  </HMI_ButtonLampTglCircleReds>
</HMI_ButtonLampTglCircleRed_Xml>
```



対象 XML	濃青	HMI_ButtonLampTglCircleBlue.xml	
イメージ		XML 要素名/属性名	説明
OFF(0)		HMI_ButtonLampTglCircleBlue	タイルを定義する要素名
ON(1)		Tag	ON の設定に使用する Tag
		Text	表示する固定ラベル
		Top	画面上からの位置(px 指定)
		Left	画面左からの位置(px 指定)
		FontSize	ラベルサイズ
		Width	表示幅
		Zindex	要素の前後関係(値が大きいかほど手前に、小さいほど奥に表示)
Xml サンプル			
<pre>&lt;HMI_ButtonLampTglCircleBlue_Xml&gt;   &lt;HMI_ButtonLampTglCircleBlues&gt;     &lt;HMI_ButtonLampTglCircleBlue Tag="Command_Start_2" Text="運転" FontSize="22px" Top="600px" Left="1000px" Width="80px" /&gt;   &lt;/HMI_ButtonLampTglCircleBlues&gt; &lt;/HMI_ButtonLampTglCircleBlue_Xml&gt;</pre>			

## 8.2.4.ボタン ON/OFF 設定 | 矩形

タイル名	種類	備考	説明
ボタン	ON/OFF 設定	形状: 矩形	Tag の値が OFF(0)の場合 Off の表示 ボタン押下で ON(1)を設定 Tag の値が ON(1)の場合 On の表示 ボタン押下で OFF(0)を設定

対象 XML	緑	HMI_ButtonLampTglRectGreen.xml	
イメージ	XML 要素名/属性名		説明
OFF(0)		HMI_ButtonLampTglRectGreen	タイルを定義する要素名
		Tag	ON の設定に使用する Tag
		Text	表示する固定ラベル
		Top	画面上からの位置(px 指定)
		Left	画面左からの位置(px 指定)
		Height	表示高さ
		FontSize	ラベルサイズ
		Width	表示幅
		Zindex	要素の前後関係(値が大きいほど手前に、小さいほど奥に表示)
ON(1)			

### Xml サンプル

```
<HMI_ButtonLampTglRectGreen_Xml>
  <HMI_ButtonLampTglRectGreens>
    <HMI_ButtonLampTglRectGreen Tag="Command_Stop" Text="停止" FontSize="22px" Top="310px" Left="130px" Width="110px" />
  </HMI_ButtonLampTglRectGreens>
</HMI_ButtonLampTglRectGreen_Xml>
```

対象 XML	赤色	HMI_ButtonLampTglRectRed.xml
イメージ	意味	備考
OFF(0)		
ON(1)		
	HMI_ButtonLampTglRectRed	タイルを定義する要素名
	Tag	ON の設定に使用する Tag
	Text	表示する固定ラベル
	Top	画面上からの位置(px 指定)
	Left	画面左からの位置(px 指定)
	Height	表示高さ
	FontSize	ラベルサイズ
	Width	表示幅
	Zindex	要素の前後関係(値が大きいほど手前に、小さいほど奥に表示)

### Xml サンプル

```
<HMI_ButtonLampTglRectRed_Xml >
  <HMI_ButtonLampTglRectReds>
    <HMI_ButtonLampTglRectRed Tag="Emergency" Text="非常停止" FontSize="22px" Top="370px" Left="130px" Width="110px" Height="40px" />
  </HMI_ButtonLampTglRectReds>
</HMI_ButtonLampTglRectRed_Xml>
```

対象 XML		濃青	HMI_ButtonLampTglRectBlue.xml	
イメージ		XML 要素名/属性名		説明
OFF(0)		HMI_ButtonLampTglRectBlue	タイルを定義する要素名	
	↕	Tag	ON の設定に使用する Tag	
ON(1)		Text	表示する固定ラベル	
		Top	画面上からの位置(px 指定)	
		Left	画面左からの位置(px 指定)	
		Height	表示高さ	
		FontSize	ラベルサイズ	
		Width	表示幅	
		Zindex	要素の前後関係(値が大きいほど手前に、小さいほど奥に表示)	

Xml サンプル

```
<HMI_ButtonLampTglRectBlue_Xml>
  <HMI_ButtonLampTglRectBlues>
    <HMI_ButtonLampTglRectBlue Tag="Command_Start" Text="運転" FontSize="22px" Top="430px" Left="130px" Width="110px" Height="40px" />
  </HMI_ButtonLampTglRectBlues>
</HMI_ButtonLampTglRectBlue_Xml>
```

## 8.2.5.ボタン リンク | 円形

タイル名	種類	備考	説明
ボタン	リンク	形状: 円	リンク先へ移動

対象 XML	緑	HMI_ButtonLinkCircleGreen.xml
イメージ	XML 要素名/属性名	説明



HMI_ButtonLinkCircleGreen	タイルを定義する要素名
Href	リンク先 URL
Text	表示する固定ラベル
Top	画面上からの位置(px 指定)
Left	画面左からの位置(px 指定)
FontSize	ラベルサイズ
Width	表示幅
Zindex	要素の前後関係(値が大きいほど手前に、小さいほど奥に表示)

### Xml サンプル

```
<HMI_ButtonLinkCircleGreen_Xml>
  <HMI_ButtonLinkCircleGreens>
    <HMI_ButtonLinkCircleGreen Href="/" Text="リンク 1" FontSize="11px" Top="310px" Left="1700px" Width="55px" />
  </HMI_ButtonLinkCircleGreens>
</HMI_ButtonLinkCircleGreen_Xml>
```


対象 XML	赤	HMI_ButtonLinkCircleRed.xml
イメージ	意味	備考



HMI_ButtonLinkCircleRed	タイルを定義する要素名
Href	リンク先 URL
Text	表示する固定ラベル
Top	画面上からの位置(px 指定)
Left	画面左からの位置(px 指定)
FontSize	ラベルサイズ
Width	表示幅
Zindex	要素の前後関係(値が大きいほど手前に、小さいほど奥に表示)


### Xml サンプル

```
<HMI_ButtonLinkCircleRed_Xml>
  <HMI_ButtonLinkCircleReds>
    <HMI_ButtonLinkCircleRed Href="/" Text="リンク 2" FontSize="11px" Top="375px" Left="1700px" Width="55px" />
  </HMI_ButtonLinkCircleReds>
</HMI_ButtonLinkCircleRed_Xml>
```

対象 XML	濃青	HMI_ButtonLinkCircleRoyalblue.xml	
イメージ		意味	備考
		HMI_ButtonLinkCircleRoyalblue	タイルを定義する要素名
		Href	リンク先 URL
		Text	表示する固定ラベル
		Top	画面上からの位置(px 指定)
		Left	画面左からの位置(px 指定)
		FontSize	ラベルサイズ
		Width	表示幅
		Zindex	要素の前後関係(値が大きいほど手前に、小さいほど奥に表示)

Xml サンプル

```
<HMI_ButtonLinkCircleRoyalblue_Xml>
  <HMI_ButtonLinkCircleRoyalblues>
    <HMI_ButtonLinkCircleRoyalblue Href="/" Text="リンク 3" FontSize="11px" Top="440px" Left="1700px" Width="55px" />
  </HMI_ButtonLinkCircleRoyalblues>
</HMI_ButtonLinkCircleRoyalblue_Xml>
```

対象 XML	オレンジ色	HMI_ButtonLinkCircleOrange.xml	
イメージ		意味	備考
		HMI_ButtonLinkCircleOrange	タイルを定義する要素名
		Href	リンク先 URL
		Text	表示する固定ラベル
		Top	画面上からの位置(px 指定)
		Left	画面左からの位置(px 指定)
		FontSize	ラベルサイズ
		Width	表示幅
		Zindex	要素の前後関係(値が大きいほど手前に、小さいほど奥に表示)

Xml サンプル

```
<HMI_ButtonLinkCircleOrange_Xml>
  <HMI_ButtonLinkCircleOranges>
    <HMI_ButtonLinkCircleOrange Href="/" Text="リンク 4" FontSize="11px" Top="505px" Left="1700px" Width="55px" />
  </HMI_ButtonLinkCircleOranges>
</HMI_ButtonLinkCircleOrange_Xml>
```

対象 XML	濃紺	HMI_ButtonLinkCircleDarkblueOver.xml	
イメージ	意味	備考	
	HMI_ButtonLinkCircleDarkblue Over	タイルを定義する要素名	
	Href	リンク先 URL	
	Text	表示する固定ラベル	
	Top	画面上からの位置(px 指定)	
	Left	画面左からの位置(px 指定)	
	FontSize	ラベルサイズ	
	Width	表示幅	
	Zindex	要素の前後関係(値が大きいほど手前に、小さいほど奥に表示)	

### Xml サンプル


```
<HMI_ButtonLinkCircleDarkblueOver_Xml>
  <HMI_ButtonLinkCircleDarkblueOvers>
    <HMI_ButtonLinkCircleDarkblueOver Href="/" Text="リンク 5" FontSize="11px" Top="570px" Left="1700px" Width="55px" />
  </HMI_ButtonLinkCircleDarkblueOvers>
</HMI_ButtonLinkCircleDarkblueOver_Xml>
```

### 備考

マウスオーバー時、色を変更します。


## 8.2.6.ボタン リンク | 矩形

タイル名	種類	備考	説明
ボタン	リンク	形状: 矩形	リンク先へ移動

対象 XML	緑色	HMI_ButtonLinkRectGreen.xml	
イメージ		XML 要素名/属性名	説明
		HMI_ButtonLinkRectGreen	タイルを定義する要素名
		Href	リンク先 URL
		Top	画面上からの位置(px 指定)
		Left	画面左からの位置(px 指定)
		FontSize	ラベルサイズ
		Height	表示高さ
		Width	表示幅
		Zindex	要素の前後関係(値が大きいほど手前に、小さいほど奥に表示)


## [Xml サンプル]

```
<HMI_ButtonLinkRectGreen_Xml>
  <HMI_ButtonLinkRectGreens>
    <HMI_ButtonLinkRectGreen Href="/" Text="リンク 1" FontSize="16px" Top="310px" Left="210px" Width="120px" Height="40px" />
  </HMI_ButtonLinkRectGreens>
</HMI_ButtonLinkRectGreen_Xml>
```

対象 XML	赤色	HMI_ButtonLinkRectRed.xml	
イメージ		意味	備考
		HMI_ButtonLinkRectRed	タイルを定義する要素名
		Href	リンク先 URL
		Top	画面上からの位置(px 指定)
		Left	画面左からの位置(px 指定)
		Height	表示高さ
		FontSize	ラベルサイズ
		Width	表示幅
		Zindex	要素の前後関係(値が大きいほど手前に、小さいほど奥に表示)


## Xml サンプル

```
<HMI_ButtonLinkRectRed_Xml>
  <HMI_ButtonLinkRectReds>
    <HMI_ButtonLinkRectRed Href="/" Text="リンク 2" FontSize="16px" Top="375px" Left="2100px" Width="120px" Height="40px" />
  </HMI_ButtonLinkRectReds>
</HMI_ButtonLinkRectRed_Xml>
```

対象 XML	濃青	HMI_ButtonLinkRectRoyalblue.xml	
イメージ		意味	備考
		HMI_ButtonLinkRectRoyalblue	タイルを定義する要素名
		Href	リンク先 URL
		Top	画面上からの位置(px 指定)
		Left	画面左からの位置(px 指定)
		Height	表示高さ
		FontSize	ラベルサイズ
		Width	表示幅
		Zindex	要素の前後関係(値が大きいほど手前に、小さいほど奥に表示)

Xml サンプル

```
<HMI_ButtonLinkRectRoyalblue_Xml>
  <HMI_ButtonLinkRectRoyalblues>
    <HMI_ButtonLinkRectRoyalblue Href="https://www.mnc.co.jp/" Text="リンク 4" FontSize="16px" Top="440px" Left="210px" Width="120px" Height="40px" />
  </HMI_ButtonLinkRectRoyalblues>
</HMI_ButtonLinkRectRoyalblue_Xml>
```

対象 XML	オレンジ	HMI_ButtonLinkRectOrange.xml	
イメージ		意味	備考
		HMI_ButtonLinkRectOrange	タイルを定義する要素名
		Href	リンク先 URL
		Top	画面上からの位置(px 指定)
		Left	画面左からの位置(px 指定)
		Height	表示高さ
		FontSize	ラベルサイズ
		Width	表示幅
		Zindex	要素の前後関係(値が大きいほど手前に、小さいほど奥に表示)

Xml サンプル

```
<HMI_ButtonLinkRectOrange_Xml>
  <HMI_ButtonLinkRectOranges>
    <HMI_ButtonLinkRectOrange Href="/" Text="リンク 4" FontSize="11px" Top="505px" Left="210px" Width="120px" Height="40px" />
  </HMI_ButtonLinkRectOranges>
</HMI_ButtonLinkRectOrange_Xml>
```



対象 XML	濃紺	HMI_ButtonLinkRectDarkblueOver.xml	
イメージ	意味	備考	
	HMI_ButtonLinkRectDarkblueOver	タイルを定義する要素名	
	Href	リンク先 URL	
	Top	画面上からの位置(px 指定)	
	Left	画面左からの位置(px 指定)	
	Height	表示高さ	
	FontSize	ラベルサイズ	
	Width	表示幅	
	Zindex	要素の前後関係(値が大き いほど手前に、小さいほど 奥に表示)	

Xml サンプル

<HMI\_ButtonLinkRectDarkblueOver\_Xml>

<HMI\_ButtonLinkRectDarkblueOvers>

<HMI\_ButtonLinkRectDarkblueOver Href="/" Text="リンク 5" FontSize="11px" Top="570px" Left="210px" Width="120px" Height="40px" />

</HMI\_ButtonLinkRectDarkblueOvers>

</HMI\_ButtonLinkRectDarkblueOver\_Xml>

## 8.3. ランプ



ランプ表示を行うタイルです:

### 8.3.1. ランプ 円形

タイル名	種類	備考	説明
ランプ		形状: 円	ON(1)時対象の色に変更 OFF(0)時はグレー
対象 XML	緑	HMI_LampCircleGreen.xml	
イメージ		XML 要素名/属性名	説明
<div>OFF(0) </div> <div>ON(1) </div>		HMI_LampCircleGreen	タイルを定義する要素名
		Tag	ON の設定に使用する Tag
		Text	表示する固定ラベル
		Top	画面上からの位置(px 指定)
		Left	画面左からの位置(px 指定)
		FontSize	ラベルサイズ
		Width	表示幅
		Zindex	要素の前後関係(値が大きいのほど手前に、小さいほど奥に表示)

Xml サンプル

```
<HMI_LampCircleGreen_Xml>
<HMI_LampCircleGreens>
<HMI_LampCircleGreen Tag="Running_01" Text="運転 1" FontSize="20px" Top="720px" Left="500px" Width="60px" />
</HMI_LampCircleGreens>
</HMI_LampCircleGreen_Xml>
```



対象 XML	赤色	HMI_ButtonCircleRed.xml	
イメージ	意味	備考	
<div> OFF(0)    ON(1)  </div>	HMI_ButtonCircleRed		タイルを定義する要素名
	Tag		ON の設定に使用する Tag
	Text		表示する固定ラベル
	Top		画面上からの位置(px 指定)
	Left		画面左からの位置(px 指定)
	FontSize		ラベルサイズ
	Width		表示幅
	Zindex		要素の前後関係(値が大きいのほど手前に、小さいほど奥に表示)

Xml サンプル

```
<HMI_ButtonCircleRed_Xml >
<HMI_ButtonCircleReds>
<HMI_ButtonCircleRed Tag="Emergency_02" Text="停止 2" FontSize="15px" Top="375px" Left="500px" Width="55px" />
</HMI_ButtonCircleReds>
</HMI_ButtonCircleRed_Xml>
```

タイル名	種類	備考	説明
ランプ		形状: 矩形	ON(1)時対象の色に変更 OFF(0)時はグレー

対象 XML	緑	HMI_LampRectGreen.xml
--------	---	-----------------------

イメージ	XML 要素名/属性名	説明
OFF(0) 	HMI_LampRectGreen	タイルを定義する要素名
	Tag	ON の設定に使用する Tag
	Text	表示する固定ラベル
	Top	画面上からの位置(px 指定)
ON(1) 	Left	画面左からの位置(px 指定)
	FontSize	ラベルサイズ
	Width	表示幅
	Zindex	要素の前後関係(値が大きいかほど手前に、小さいほど奥に表示)

#### Xml サンプル

```
<HMI_LampRectGreen_Xml>
  <HMI_LampRectGreens>
    <HMI_LampRectGreen Tag="Running_01" Text="運転 1" FontSize="20px" Top="720px" Left="500px" Width="60px" />
  </HMI_LampRectGreens>
</HMI_LampRectGreen_Xml>
```

対象 XML	赤色	HMI_ButtonRectRed.xml
--------	----	-----------------------

イメージ	意味	備考
OFF(0)		
	↕	
ON(1)		

HMI_ButtonRectRed	タイルを定義する要素名
Tag	ON の設定に使用する Tag
Text	表示する固定ラベル
Top	画面上からの位置(px 指定)
Left	画面左からの位置(px 指定)
FontSize	ラベルサイズ
Width	表示幅
Zindex	要素の前後関係(値が大きいかほど手前に、小さいほど奥に表示)

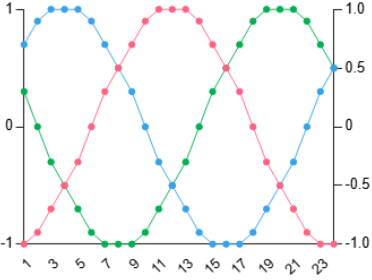
#### Xml サンプル

```
<HMI_ButtonRectRed_Xml >
  <HMI_ButtonRectReds>
    <HMI_ButtonRectRed Tag="Emergency_02" Text="停止 2" FontSize="15px" Top="375px" Left="500px" Width="55px" />
  </HMI_ButtonRectReds>
</HMI_ButtonRectRed_Xml>
```

## 8.4. チャート

チャート表示を行うタイルです:

### 8.4.1. チャート 折れ線グラフ

タイル名	種類	備考	説明
チャート	折れ線グラフ	折れ線グラフを表示 データ種類は最大 3 種類、全データ全タグ指定 1 軸 or 2 軸表示	
対象 XML	背景 白	HMI_ChartLineWhite.xml	
イメージ	XML 要素名/属性名		説明
	HMI_ChartLineWhites		タイルを定義する要素名
	Top		画面上からの位置(px 指定)
	Left		画面左からの位置(px 指定)
	Width		表示幅(px 指定)
	ChartId		Chart を識別するユニーク ID※
	ChartOpt_Scales_Y1_Ticks_StepSize		Y 軸(左側) 目盛のステップ幅
	ChartOpt_Scales_Y2_Ticks_StepSize		Y 軸(右側) 目盛のステップ幅
	ChartOpt_Scales_Y1_Ticks_Display		Y 軸(左側) 目盛 表示 ON(true)/OFF(false)
	ChartOpt_Scales_Y2_Ticks_Display		Y 軸(右側) 目盛 表示 ON(true)/OFF(false)
	ChartOpt_Plugins_Legend_Display		凡例表示 ON(true)/OFF(false)
	ChartOpt_Plugins_Legend_Position		凡例表示位置 'top'(上部), 'left'(左側), 'bottom'(下部), 'right'(右側), 'chartArea'(チャ ート領域) のいずれか
	ChartData1_Label		データ 1 名前(凡例表示用)
	ChartData2_Label		データ 2 名前(凡例表示用)
	ChartData3_Label		データ 3 名前(凡例表示用)
データ部	ChartDataSets		データを定義する要素名
	ChartDataSet		データを定義する要素名 X 軸 1 個 Y 軸最大 3 個まで指定可能
	Axis		軸名 x, y1, y2 の何れかをシチエ
	IsTag		タグの値を参照するか否かの指 定(true : タグ名指定、false:固定 値指定)
	Digit		有効桁数(表示桁数)
データ詳細部	Data		1 データを定義する要素名 値: タグ名 or 固定値 タグ名の場合はタグの値を表示 します。

## Xml サンプル

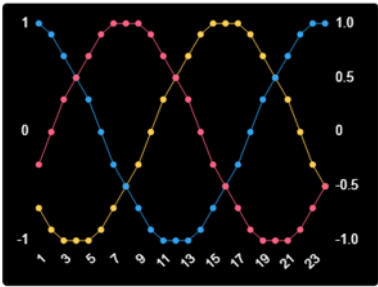
```
<HMI_ChartLineWhite_Xml>
<HMI_ChartLineWhites>
  <HMI_ChartLineWhite Top="100px" Left="100px" Width="500px" Height="300px" ChartId="Chart2"
    ChartOpt_Scales_Y1_Ticks_StepSize="0.2" ChartOpt_Scales_Y2_Ticks_Display="false" ChartOpt_Plugins_Legend_Display="true"
    ChartOpt_Plugins_Legend_Position="right" ChartData1_Label="U 相" ChartData2_Label="V 相" ChartData3_Label="W 相">
    <ChartDataSets>
      <ChartDataSet Axis="x" IsTag="true" Digit="1">
        <Data>ChartDataTagX-1</Data>
        <Data>ChartDataTagX-2</Data>
        <Data>ChartDataTagX-3</Data>
        <Data>ChartDataTagX-4</Data>
        <Data>ChartDataTagX-5</Data>
        <Data>ChartDataTagX-6</Data>
        <Data>ChartDataTagX-7</Data>
        <Data>ChartDataTagX-8</Data>
        <Data>ChartDataTagX-9</Data>
        <Data>ChartDataTagX-10</Data>
      </ChartDataSet>
      <ChartDataSet Axis="y1" IsTag="true" Digit="1">
        <Data>ChartDataTag1-1</Data>
        <Data>ChartDataTag1-2</Data>
        <Data>ChartDataTag1-3</Data>
        <Data>ChartDataTag1-4</Data>
        <Data>ChartDataTag1-5</Data>
        <Data>ChartDataTag1-6</Data>
        <Data>ChartDataTag1-7</Data>
        <Data>ChartDataTag1-8</Data>
        <Data>ChartDataTag1-9</Data>
        <Data>ChartDataTag1-10</Data>
      </ChartDataSet>
      <ChartDataSet Axis="y1" IsTag="true" Digit="1">
        <Data>ChartDataTag2-1</Data>
        <Data>ChartDataTag2-2</Data>
        <Data>ChartDataTag2-3</Data>
        <Data>ChartDataTag2-4</Data>
        <Data>ChartDataTag2-5</Data>
        <Data>ChartDataTag2-6</Data>
        <Data>ChartDataTag2-7</Data>
        <Data>ChartDataTag2-8</Data>
        <Data>ChartDataTag2-9</Data>
        <Data>ChartDataTag2-10</Data>
      </ChartDataSet>
      <ChartDataSet Axis="y1" IsTag="true" Digit="1">
        <Data>ChartDataTag3-1</Data>
        <Data>ChartDataTag3-2</Data>
        <Data>ChartDataTag3-3</Data>
        <Data>ChartDataTag3-4</Data>
        <Data>ChartDataTag3-5</Data>
        <Data>ChartDataTag3-6</Data>
        <Data>ChartDataTag3-7</Data>
        <Data>ChartDataTag3-8</Data>
        <Data>ChartDataTag3-9</Data>
        <Data>ChartDataTag3-10</Data>
      </ChartDataSet>
    </ChartDataSets>
  </HMI_ChartLineWhite>
</HMI_ChartLineWhites>
</HMI_ChartLineWhite_Xml>
```

X 軸に指定する Tag 名を指定  
上から順に X 軸左 1～x 番目のメモリ値を指定

Y 軸に指定する Tag 名を指定  
上から順に X 軸左 1～x 番目のデータ値を指定

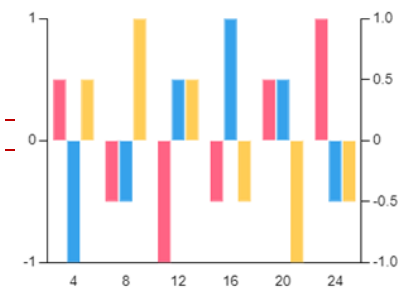
## ※備考

- ChartId がチャートタイトル内で重複した場合、Chart は画面に表示されません。
- 本タイルは全てタグのデータを表示します。時系列データのようなグラフにする場合は、タグの値を逐次変更してください。

対象 XML	背景 黒	HMI_ChartLineBlack.xml
イメージ	XML 要素名/属性名	説明
	HMI_ChartLineBlacks	タイルを定義する要素名
	その他 HMI_ChartLineWhite と同様の為省略 (HMI_ChartLineWhite を参照ください)	
<p>Xml サンプル</p> <pre> &lt;HMI_ChartLineBlack_Xml &gt;   &lt;HMI_ChartLineBlacks &gt;     &lt;HMI_ChartLineBlack Top="100px" Left="100px" Width="500px" Height="300px" ChartId="Chart2"       ChartOpt_Scales_Y1_Ticks_StepSize="0.2" ChartOpt_Scales_Y2_Ticks_Display="false" ChartOpt_Plugins_Legend_Display="true"       ChartOpt_Plugins_Legend_Position="right" ChartData1_Label="U 相" ChartData2_Label="V 相" ChartData3_Label="W 相"&gt;       &lt;ChartDataSets&gt;         . . . (HMI_ChartLineWhite と同様の為省略)       &lt;/ChartDataSets&gt;     &lt;/HMI_ChartLineBlack&gt;   &lt;/HMI_ChartLineBlacks&gt; &lt;/HMI_ChartLineBlack_Xml&gt; </pre>		
<p>※備考</p> <ul style="list-style-type: none"> <li>ChartId がチャートタイトル内で重複した場合、Chart は画面に表示されません。</li> <li>本タイルは全てタグのデータを表示します。時系列データのようなグラフにする場合は、タグの値を逐次変更してください。</li> </ul>		

## 8.4.2.チャート 棒グラフ

タイル名	種類	備考	説明
チャート	棒グラフ		棒グラフを表示 データ種類は最大 3 種類、全データ全タグ指定 1 軸 or 2 軸表示

対象 XML	背景 白	HMI_ChartBarWhite.xml
イメージ	XML 要素名/属性名	説明
	<p>HMI_ChartBarWhites</p> <p>その他 HMI_ChartLineWhite と同様の為省略 (HMI_ChartLineWhite を参照ください)</p>	<p>タイルを定義する要素名</p>

### Xml サンプル

```
<HMI_ChartBarWhite_Xml>
<HMI_ChartBarWhites>
  <HMI_ChartBarWhite Top="2000px" Left="700px" Width="400px" Height="300px" ChartId="Chart3"
    ChartOpt_Scales_Y1_Ticks_StepSize="5" ChartOpt_Scales_Y2_Ticks_StepSize="0.5">
    <ChartDataSets>
      <ChartDataSet Axis="x" IsTag="false">
        <Data>4</Data>
        <Data>8</Data>
        <Data>12</Data>
        <Data>16</Data>
        <Data>20</Data>
        <Data>24</Data>
      </ChartDataSet>
      <ChartDataSet Axis="y1" IsTag="true" Digit="1">
        <Data>ChartDataTag1-4</Data>
        <Data>ChartDataTag1-8</Data>
        <Data>ChartDataTag1-12</Data>
        <Data>ChartDataTag1-16</Data>
        <Data>ChartDataTag1-20</Data>
        <Data>ChartDataTag1-24</Data>
      </ChartDataSet>
      <ChartDataSet Axis="y1" IsTag="true" Digit="1">
        <Data>ChartDataTag2-4</Data>
        <Data>ChartDataTag2-8</Data>
        <Data>ChartDataTag2-12</Data>
        <Data>ChartDataTag2-16</Data>
        <Data>ChartDataTag2-20</Data>
        <Data>ChartDataTag2-24</Data>
      </ChartDataSet>
      <ChartDataSet Axis="y2" IsTag="true" Digit="1">
        <Data>ChartDataTag3-4</Data>
        <Data>ChartDataTag3-8</Data>
        <Data>ChartDataTag3-12</Data>
        <Data>ChartDataTag3-16</Data>
        <Data>ChartDataTag3-20</Data>
        <Data>ChartDataTag3-24</Data>
      </ChartDataSet>
    </ChartDataSets>
  </HMI_ChartLineWhite>
</HMI_ChartLineWhites>
</HMI_ChartLineWhite_Xml>
```

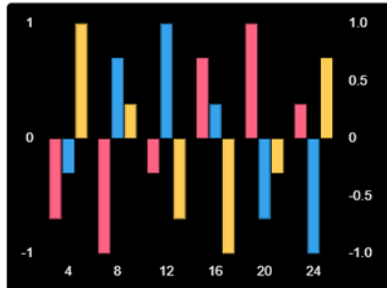
<HMI\_ChartLineWhite と同様>  
X 軸に指定する Tag 名を指定  
上から順に X 軸左 1~x 番目のメモリ値を指定

### ※備考

- ChartId がチャートタイル内で重複した場合、Chart は画面に表示されません。
- 本タイルは全てタグのデータを表示します。時系列データのようなグラフにする場合は、タグの値を逐次変更し

てください。

対象 XML	背景 黒	HMI_ChartBarBlack.xml
イメージ	XML 要素名/属性名	説明



HMI\_ChartBarBlacks

タイルを定義する要素名

その他 HMI\_ChartLineWhite と同様の為省略  
(HMI\_ChartLineWhite を参照ください)

#### Xml サンプル

```
<HMI_ChartBarBlack_Xml>
  <HMI_ChartBarBlacks >
    <HMI_ChartBarBlack Top="1700px" Left="700px" Width="400px" Height="300px" ChartId="Chart1"
      ChartOpt_Scales_Y1_Ticks_StepSize="5" ChartOpt_Scales_Y2_Ticks_StepSize="0.5">
      <ChartDataSets>
        . . . (HMI_ChartLineWhite と同様の為省略)
      </ChartDataSets>
    </HMI_ChartBarBlack>
  </HMI_ChartBarBlacks>
</HMI_ChartBarBlack_Xml>
```

#### ※備考

- ChartId がチャートタイトル内で重複した場合、Chart は画面に表示されません。
- 本タイルは全てタグのデータを表示します。時系列データのようなグラフにする場合は、タグの値を逐次変更し  
てください。



## 8.5. ログ表示

ログ表示を行うタイルです:

### 8.5.1. ログ表示 テーブル形式

タイル名	種類	備考	説明
ログ表示	テーブル形式	テーブル形式 ログを表示 全データ全タグ指定	
対象 XML	背景 白	HMI_LogViewTableWhite.xml	
イメージ	XML 要素名/属性名		説明
	HMI_LogViewTableWhite		タイルを定義する要素名
	Top		画面上からの位置(px 指定)
	Left		画面左からの位置(px 指定)
	FontSize		文字サイズ(px 指定)
	Width		表示幅(px 指定)
	Height		表示高さ(px 指定)
	LogViewTableId		Table を識別するユニーク ID※
	Border		枠線の幅を指定
	TableData		データを定義する要素名
	Column		データを定義する要素名 1 列ずつ指定可能
データ部	Header		1 行目に表示する内容
	IsTag		タグの値を参照するか否かの指定(true : タグ名指定、false:固定値指定)
	Digit		有効桁数(表示桁数)
	ColWidth		テーブルの列の幅(px 指定)
	Row		1 データを定義する要素名 値 : タグ名 or 固定値 タグ名の場合はタグの値を表示します。
データ詳細部			

## Xml サンプル

```
<HMI_LogViewTableWhite_Xml >
  <HMI_LogViewTableWhites >
    <HMI_LogViewTableWhite Top="2800px" Left="400px" Width="600px" Height="200px" LogViewTableId="Table1" FontSize="20px"
      Border="2px solid #7F8591" >
      <TableData >
        <Column Header="No." Digit="0" IsTag="false" ColWidth="80px">
          <Row>1</Row>
          <Row>2</Row>
          <Row>3</Row>
          <Row>4</Row>
          <Row>5</Row>
          <Row>6</Row>
          <Row>7</Row>
          <Row>8</Row>
          <Row>9</Row>
          <Row>10</Row>
        </Column>
        <Column Header="日時" Digit="0" IsTag="true" ColWidth="120px">
          <Row>LogDataTag1-1</Row>
          <Row>LogDataTag1-2</Row>
          <Row>LogDataTag1-3</Row>
          <Row>LogDataTag1-4</Row>
          <Row>LogDataTag1-5</Row>
          <Row>LogDataTag1-6</Row>
          <Row>LogDataTag1-7</Row>
          <Row>LogDataTag1-8</Row>
          <Row>LogDataTag1-9</Row>
          <Row>LogDataTag1-10</Row>
        </Column>
        <Column Header="信号名" ColWidth="100px">
          <Row>LogDataTag2-1</Row>
          <Row>LogDataTag2-2</Row>
          <Row>LogDataTag2-3</Row>
          <Row>LogDataTag2-4</Row>
          <Row>LogDataTag2-5</Row>
          <Row>LogDataTag2-6</Row>
          <Row>LogDataTag2-7</Row>
          <Row>LogDataTag2-8</Row>
          <Row>LogDataTag2-9</Row>
          <Row>LogDataTag2-10</Row>
        </Column>
        <Column Header="イベント">
          <Row>LogDataTag3-1</Row>
          <Row>LogDataTag3-2</Row>
          <Row>LogDataTag3-3</Row>
          <Row>LogDataTag3-4</Row>
          <Row>LogDataTag3-5</Row>
          <Row>LogDataTag3-6</Row>
          <Row>LogDataTag3-7</Row>
          <Row>LogDataTag3-8</Row>
          <Row>LogDataTag3-9</Row>
          <Row>LogDataTag3-10</Row>
        </Column>
      </TableData>
    </HMI_LogViewTableWhite>
  </HMI_LogViewTableWhites>
</HMI_LogViewTableWhite_Xml>
```

1 列目に指定する固定値を指定  
上から順に 1～x 番目のメモリ値を指定

2 列目に指定する Tag 名を指定  
上から順に 1～x 番目のデータ値を指定

### ※備考

- LogViewTableId がログ表示内で重複した場合、LogViewTable は画面に表示されません。
- 本タイルは全てタグのデータを表示します。

対象 XML			背景 青	HMI_LogViewTableBlue.xml													
イメージ			XML 要素名/属性名		説明												
<table><thead><tr><th>日時</th><th>信号名</th><th>イベント</th></tr></thead><tbody><tr><td>2024-11-19 18:06:22</td><td>運転</td><td>ON</td></tr><tr><td>2024-11-19 18:06:19</td><td>緊急停止</td><td>OFF</td></tr><tr><td>2024-11-19 18:06:15</td><td>緊急停止</td><td>ON</td></tr></tbody></table>			日時	信号名	イベント	2024-11-19 18:06:22	運転	ON	2024-11-19 18:06:19	緊急停止	OFF	2024-11-19 18:06:15	緊急停止	ON	HMI_LogViewTableBlue		タイルを定義する要素名
			日時	信号名	イベント												
			2024-11-19 18:06:22	運転	ON												
			2024-11-19 18:06:19	緊急停止	OFF												
			2024-11-19 18:06:15	緊急停止	ON												
			HeadForeColor		1 行目の文字の色												
HeadBackColor		1 行目の背景色															
BodyForeColor		2 行目以上の文字の色															
BodyBackColor		2 行目以上の背景色															
HoverColor		2 行目以上のマウスオーバー時の背景色															

その他 HMI\_LogViewTableWhite と同様の為省略  
(HMI\_LogViewTableWhite を参照ください)

#### Xml サンプル

```
<HMI_LogViewTableBlue_Xml>
  <HMI_LogViewTableBlues >
    <HMI_LogViewTableBlue Top="100px" Left="100px" Width="500px" Height="300px" LogViewTableId="Table1" FontSize="20px"
    Border="2px solid #7F8591" HeadForeColor="#F0F0F0" HeadBackColor="#1D3854" BodyForeColor="#F0F0F0" BodyBackColor="#2D5680"
    HoverColor="#4e6e9b">
      <TableData>
        . . . (HMI_LogViewTableWhite と同様の為省略)
      </TableData>
    </ChartDataSets>
  </HMI_LogViewTableBlue>
</HMI_LogViewTableBlues>
</HMI_LogViewTableBlue_Xml>
```

#### ※備考

- LogViewTableId がログ表示内で重複した場合、LogViewTable は画面に表示されません。
- 本タイルは全てタグのデータを表示します。

## 8.6. インナーフレーム

インナーフレーム処理を行うタイルです:

タイル名	種類	備考	説明
------	----	----	----

インナーフレーム

HTML の IFrame 処理

対象 XML	背景	タイル名
	グレー	HMI_FrameGray.xml

イメージ

XML 要素名/属性名

説明



HMI_FrameGray	タイルを定義する要素名
Source	ON の設定に使用する Tag
Id	表示する固定ラベル
Border	フレームの境界線(0:非表示、1:表示)
Transform	回転、拡大、縮小、移動、傾斜などの変形 (scale(値):拡大縮小、rotate(値 deg):時計回りに回転、translate(値 px, 値 px):移動)
Position	要素の位置(static: 初期値、relative: 現在の位置を基準に相対的な位置、absolute: 親要素を基準に絶対的な位置、fixed: 画面のきまった位置に固定)
IsTag	Source でタグの値を参照するか否かの指定(true :タグ名指定、false:固定値指定)
Top	画面上からの位置(px 指定)
Left	画面左からの位置(px 指定)
Height	表示高さ
Width	表示幅

Xml サンプル

```
< HMI_FrameGray_Xml>
  < HMI_FrameGrays>
    < HMI_FrameGray Source="IFrameDataTag" IsTag="true" Id="ContentPlaceHolder1_iframe2" Transform="scale(1.0)"
      Position="absolute" Top="120px" Left="690px" Height="300px" Width="400px" Border="1" />
  </ HMI_FrameGrays>
</ HMI_FrameGray_Xml>
```

## 8.7. テキストボックス

テキストボックスの表示・処理を行うタイルです:

タイル名	種類	備考	説明
テキストボックス	シングル		テキストボックスと設定ボタンが対となったタイル

対象 XML

青

HMI\_TextBoxSingleWhiteBlue.xml

イメージ

XML 要素名/属性名

説明

設定

HMI_TextBoxSingleWhiteBlue	タイルを定義する要素名
Tag	ON の設定に使用する Tag
Text	表示する固定ラベル
Top	画面上からの位置(px 指定)
Left	画面左からの位置(px 指定)
FontSize	ラベルサイズ
InputDataType	データの型(1:Boolean、2:SByte、3:Byte、4:Int16、5:UInt16、6: Int32、7:UInt32、8:Int64、9:UInt64、10:Float、11:Double、12:String)
InputWidth	入力の表示幅
ButtonWidth	ボタンの表示幅
Height	入力とボタンの表示縦幅
Zindex	要素の前後関係(値が大きいかほど手前に、小さいほど奥に表示)



Xml サンプル

```
< HMI_TextBoxSingleWhiteBlue_Xml >
< HMI_TextBoxSingleWhiteBlues >
< HMI_TextBoxSingleWhiteBlue Tag="TextBoxTest_Double" FontSize="20px" Top="700px" Left="800px" InputWidth="200px"
InputDataType="11" ButtonWidth="110px" Height="40px" Text="設定" Zindex="1"/>
</ HMI_TextBoxSingleWhiteBlues >
</ HMI_TextBoxSingleWhiteBlue_Xml >
```

対象 XML	緑	HMI_TextBoxSingleWhiteGreen.xml	
イメージ	XML 要素名/属性名		説明
	HMI_TextBoxSingleWhiteGreen		タイルを定義する要素名
			その他 HMI_TextBoxSingleWhiteBlue と同様の為省略 (HMI_TextBoxSingleWhiteBlue を参照ください)

Xml サンプル

```
<HMI_TextBoxSingleWhiteGreen_Xml>
<HMI_TextBoxSingleWhiteGreens>
<HMI_TextBoxSingleWhiteGreen Tag="TextBoxTest_Double" FontSize="20px" Top="775px" Left="800px" InputWidth="200px"
InputDataType="11" ButtonWidth="110px" Height="40px" Text="設定" Zindex="1"/>
</HMI_TextBoxSingleWhiteGreens>
</HMI_TextBoxSingleWhiteGreen_Xml>
```

対象 XML	オレンジ	HMI_TextBoxSingleWhiteOrange.xml	
イメージ	XML 要素名/属性名		説明
 	HMI_TextBoxSingleWhiteOrange		タイルを定義する要素名
その他 HMI_TextBoxSingleWhiteBlue と同様の為省略 (HMI_TextBoxSingleWhiteBlue を参照ください)			

### Xml サンプル

```
<HMI_TextBoxSingleWhiteOrange_Xml>
  <HMI_TextBoxSingleWhiteOranges>
    <HMI_TextBoxSingleWhiteOrange Tag="TextBoxTest_Double" FontSize="20px" Top="850px" Left="800px" InputWidth="200px"
    InputDataType="11" ButtonWidth="110px" Height="40px" Text="設定" Zindex="1"/>
  </HMI_TextBoxSingleWhiteOranges>
</HMI_TextBoxSingleWhiteOrange_Xml>
```

## 9. 設定サンプル

ここでは各タイルの動作を確認可能な設定について説明します。サーバー部、Base 部の設定に関してはインストール時のデフォルト設定で動作します。

### 1) サーバー部の設定(HMI\_Server.xml)

表 6.サーバー部設定情報(ECI)

項目	設定値
.AutoRun	True (起動時に開始)

### 2) Base 部で表示に使用するタグの設定

表 7.サーバー部設定情報(ECI) タグ登録設定

タグ名	参照元タイル名	種類	備考
Emergency	ボタン	ON(1)/OFF(0)設定ボタン 赤色	非常停止
Command_Start	ボタン	ON(1)/OFF(0)設定ボタン 青色	運転
Command_Stop	ボタン	ON(1)/OFF(0)設定ボタン 緑色	停止
LogDataTag1-1~10	ログ表示	時間表示 日時 2~11 行目	日時
LogDataTag2-1~10	ログ表示	文字表示 オブジェクト名 2~11 行目	オブジェクト名
LogDataTag3-1~10	ログ表示	文字表示 イベント 2~11 行目	イベント
ChartDataTagX-1~24	チャート	値表示 X 軸の目盛り	時間
ChartDataTag1-1~24	チャート	データ表示 ピンク色	U 相
ChartDataTag2-1~24	チャート	データ値表示 水色	V 相
ChartDataTag3-1~24	チャート	データ値表示 緑色	W 相
IFrameDataTag	インナーフレーム	画像表示	
TextBoxTest_Double	テキストボックス	入力 設定ボタン 青色 緑色 オレンジ色	
Temp_01	ラベル	データ表示 黒色 赤色	温度
Running_01	ランプ	ON(1)/OFF(0)	運転 1
Running_02	ランプ	ON(1)/OFF(0)	運転 2

### 3) Base 部で使用するタイルとタイル定義ファイル

表 8.Base 部タイル定義ファイル

タイル名	種類	タイル定義ファイル名
ラベル	固定ラベル / 黒色	HMI_LabelFixBlack.xml
	固定ラベル / 赤色	HMI_LabelFixRed.xml
	タグラベル / 黒色	HMI_LabelTagBlack.xml
	タグラベル / 赤色	HMI_LabelTagRed.xml
ボタン	ON 設定 / 円形 / 緑色	HMI_ButtonCircleGreen.xml
	ON 設定 / 円形 / 赤色	HMI_ButtonCircleRed.xml
	ON 設定 / 矩形 / 緑色	HMI_ButtonRectGreen.xml
	ON 設定 / 矩形 / 赤色	HMI_ButtonRectRed.xml
	ON 設定 / 矩形 / 濃青色	HMI_ButtonRectRoyalblue.xml

タイル名	種類	タイル定義ファイル名
	ON 設定 / 矩形 / オレンジ色	HMI_ButtonRectOrange.xml
	ON 設定 / 矩形 / 濃紺色	HMI_ButtonRectDarkblueOver.xml
	ON/OFF 設定 / 矩形 / 緑色	HMI_ButtonLampTglRectGreen.xml
	ON/OFF 設定 / 矩形 / 赤色	HMI_ButtonLampTglRectRed.xml
	ON/OFF 設定 / 矩形 / 濃青色	HMI_ButtonLampTglRectBlue.xml
ランプ	円形 / 緑色	HMI_LampCircleGreen.xml
	円形 / 赤色	HMI_LampCircleRed.xml
	矩形 / 緑色	HMI_LampRectGreen.xml
	矩形 / 赤色	HMI_LampRectRed.xml
チャート	折れ線グラフ / 白	HMI_ChartLineWhite.xml
	折れ線グラフ / 黒	HMI_ChartLineBlack.xml
	棒グラフ / 白	HMI_ChartBarWhite.xml
	棒グラフ / 黒	HMI_ChartBarBlack.xml
ログ表示	テーブル形式 / 白	HMI_LogViewTableWhite.xml
	テーブル形式 / 青	HMI_LogViewTableBlue.xml
インナーフレーム	グレー	HMI_FrameGray.xml
テキストボックス	シングル / 青	HMI_TextBoxSingleWhiteBlue.xml
	シングル / 緑	HMI_TextBoxSingleWhiteGreen.xml
	シングル / オレンジ	HMI_TextBoxSingleWhiteOrange.xml



# 10. 動作確認

## 10.1. 動作確認手順

### 10.1.1. サーバー部動作確認手順

RT-edge ソフトウェアを起動し、HMI コンテナ が正常に起動することを確認します：

- 1) RT-Edge ソフトウェア(C:\RTEdge\EgBoot.exe)を開始します。
- 2) IIS にて対象のサイトを開始します。
- 3) RT-edge オブジェクトブラウザ(C:\RTEdge\EgBrow.exe)を起動します。



RT-edge 起動直後に RT-edge オブジェクトブラウザを起動すると初期化中の為、想定されるタグが表示されない場合があります。一度 RT-edge オブジェクトブラウザを終了し、再度起動させてください。

- 4) RT-edge コンテナ設定情報(ECI)で定義したタグが表示されていることを確認します。

RT-edge Object Browser

ファイル				
Tags	Containers	Collectors	Datasets	Mailboxes
Name	Current Value	Type	Source	Commer
SERVICE.HMI_Serv.Error	False (0)	bool		
SERVICE.HMI_Serv.Live	000038e1 (14561)	UInt32		
SERVICE.HMI_Serv.Run	True (1)	bool		
SERVICE.HMI_Serv.Status	01 (1)	byte		

図 13. コンテナ生成タグイメージ

### 10.1.2. Base 部動作確認手順

IIS で動作する Base 部を起動し、Web ブラウザで画面が正常に表示・動作する事を確認します：

- 1) [4.3. IIS の設定]を参照し、Base 部を配置し、開始します。
- 2) Web ブラウザで対象のページを参照します。
- 3) 定義したタイルが表示され、タグ情報の読み込み・書き込みが可能なことを確認します。異常時はタグの値が表示されません。

## 10.2. サービスインジケータタグ

動作確認で必要となるサービスインジケータタグについて説明します:

表 9. サービスインジケータ共通タグ一覧

ステータスインジケータ Tag 名	備考
SERVICE.<サービス名>.Status	現在のサービス起動状態を示します
SERVICE.<サービス名>.Error	現在のサービスエラー状態を示します
SERVICE.<サービス名>.Run	現在のデータタグ参照・更新動作の状態を示します
SERVICE.<サービス名>.Live	サービスが健全であることを示すカウンタ

## 10.3. RT-edge タグデータの妥当性について

タグリンクされた RT-edge タグのデータは、サービスインジケータが以下の状態になっている時、妥当であると判断出来ます。

1. SERVICE.<サービス名>.Error = FALSE (エラーが発生していない)
2. SERVICE.<サービス名>.Run = TRUE (動作が開始されている)

上記の状態になっていない場合には、何らかの問題が発生しています。 詳細は、以下のトラブルシューティングをご参照ください。

「サービスインジケータの.Error が true になっています。」

「サービスインジケータの.Run が false になっています。」

# 11. カスタムタイルの作成

標準で用意しているタイルのほかに、一からカスタマイズしたタイルを作成することができます。

## 11.1. 開発環境

タイルを開発する環境(コンパイラ・リンカ環境)

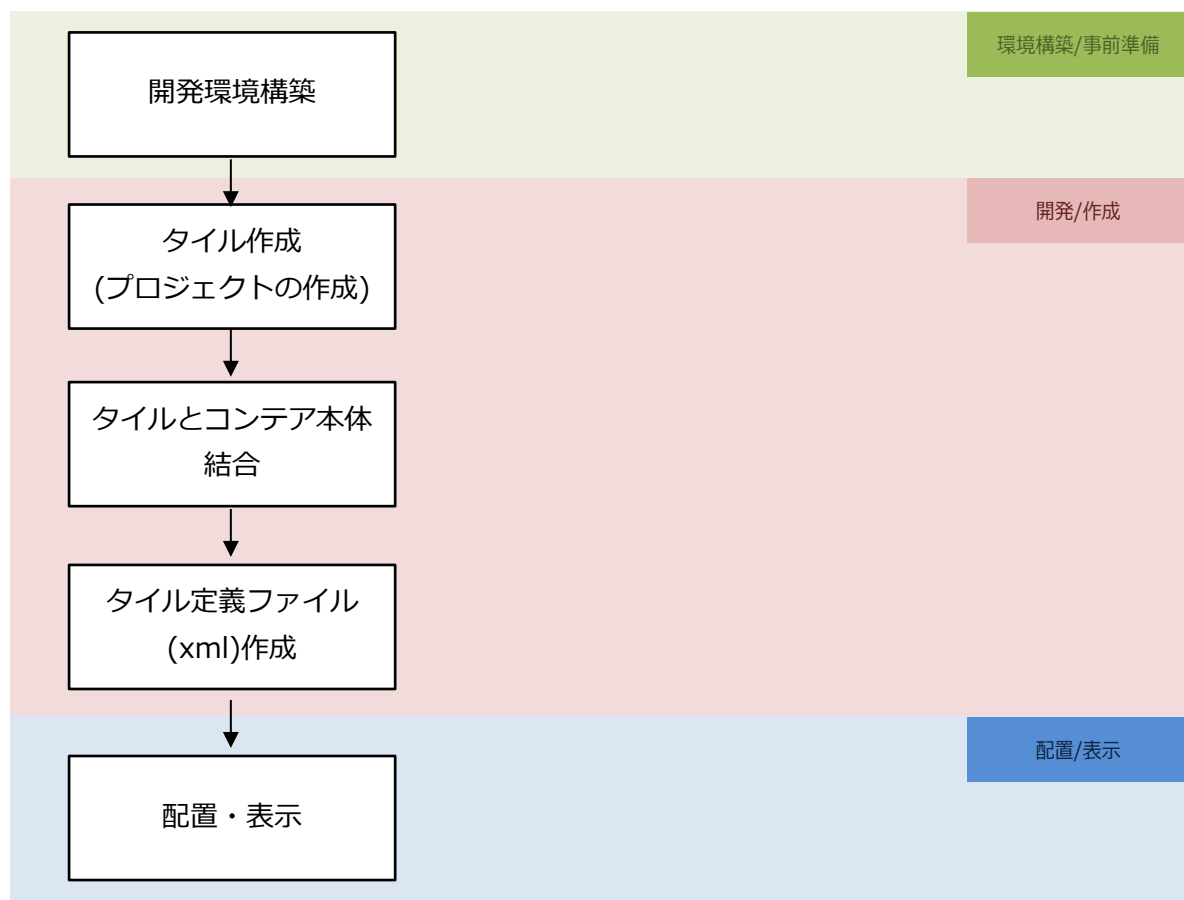
- ・ Visual Studio 2022
- ・ .NET 8



HMI コンテナ/タイルは、Blazor Server アプリケーション、Razor クラスライブラリを使用しています。Visual Studio インストール時に対象のコンポーネントをインストールして下さい。

## 11.2. 作成から運用までの流れ

タイルの作成から運用までの流れは次のようになります。



## 11.3. タイル作成例(電源ボタン作成)

例として以下の仕様のタイルを作成する

- 1) 電源のマークをしたボタンを作成する。
- 2) ボタンが押されたら、タイル定義ファイル(xml)で指定のタグの値に 1(ON)/0(OFF)を設定する。値は割り当てたタグの値が 0(OFF)の場合は 1(ON)を、1(ON)の場合は 0(OFF)を設定する。
- 3) 値を読み込み、電源ボタンの見た目を更新する

### 手順 1. 開発環境構築

Visual Studio 2022 と .NET 8 をインストールします。

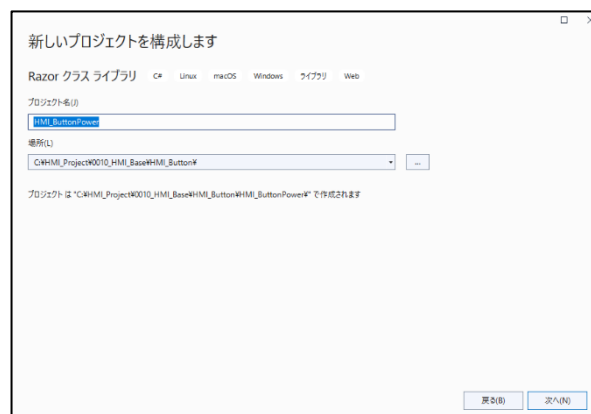
既に作成済みの場合は手順 2 に進みます。

### 手順 2. タイル作成(新しいプロジェクトの作成)

- ① インストール済みの HMI\_Base.sln を起動します。
- ② 新規でプロジェクトを追加します。例として、「HMI\_Button」フォルダ以下に作成します。ソリューションエクスプローラ上で「HMI\_Button」フォルダを選択し、コンテキストメニューより「新しいプロジェクト」を選択します。

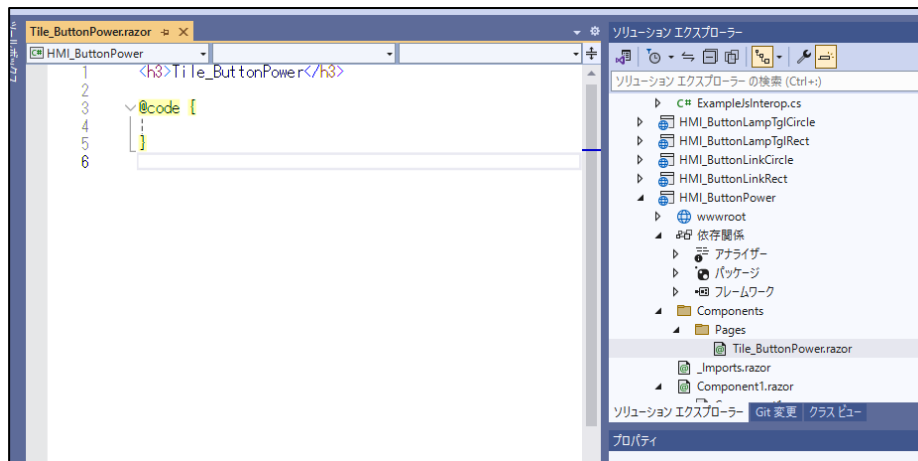


- ③ テンプレートに、「Razor クラスライブラリ」を選択後、ウィザードに従ってプロジェクトを作成します。

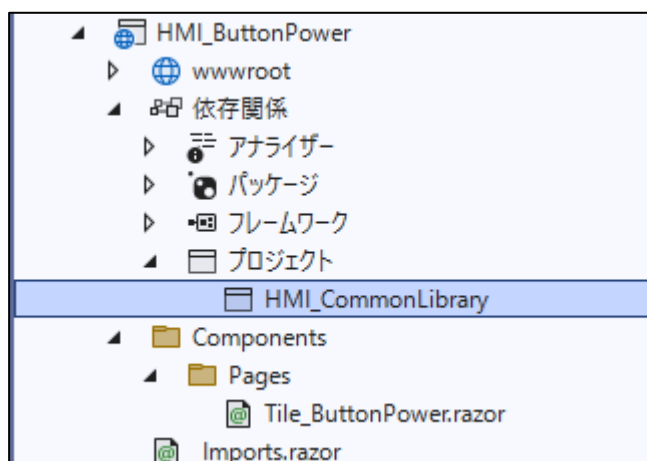


- ④ 作成後、「Components/Pages」フォルダを作成し、「Pages」以下に「Razor コンポーネント」を作成します。

例では、「Tile\_ButtonPower.razor」という名前で作成しています。



- ⑤ 「依存関係」に「プロジェクト参照の追加」から「HMI\_CommonLibrary」を追加します。  
(内部で使用するため)



- ⑥ 「Tile\_ButtonPower.razor」を実装します。以下、電源ボタンの作成例です。
- @using HMI\_CommonLibrary を記載
  - TileValue が "1" の時 ON のマークに、"0"の時 OFF のマークに変更
  - ボタンスタイルは css ファイルとして外に出しても良いが、今回は razor ファイル内に記載
  - TileButtonType = 2 は、値が 1 の場合 0 を、0 の場合 1 に変更する定義
  - ボタンクリック時の動作は別ファイルで記載(本体 DynamicPage.razor)

## &lt;Tile\_ButtonPower.razor&gt;

```

@using HMI_CommonLibrary
<!-- 電源ボタン top/left は タイル定義ファイル xml 値参照 -->
<div style="position:absolute; top:@TileTop; left:@TileLeft; z-index: @TileZindex;"
    <button @onclick="ButtonClick">
        @if (TileValue == "1")
        {
            <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 100 100" width="100" height="100">
                <!-- 背景の青いグラデーション -->
                <defs>
                    <radialGradient id="blueGradient" cx="50%" cy="50%" r="50%">
                        <stop offset="0%" stop-color="#70c1ff" />
                        <stop offset="100%" stop-color="#007bff" />
                    </radialGradient>
                </defs>
                <circle cx="50" cy="50" r="45" fill="url(#blueGradient)" />
                <!-- 白い電源マーク -->
                <circle cx="50" cy="50" r="30" stroke="white" stroke-width="8" fill="none" />
                <line x1="50" y1="10" x2="50" y2="40" stroke="white" stroke-width="8" />
            </svg>
        }
        else
        {
            <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 100 100" width="100" height="100">
                <!-- 背景のグレーのグラデーション -->
                <defs>
                    <radialGradient id="grayGradient" cx="50%" cy="50%" r="50%">
                        <stop offset="0%" stop-color="#cccccc" />
                        <stop offset="100%" stop-color="#666666" />
                    </radialGradient>
                </defs>
                <circle cx="50" cy="50" r="45" fill="url(#grayGradient)" />
                <!-- 白い電源マーク -->
                <circle cx="50" cy="50" r="30" stroke="white" stroke-width="8" fill="none" />
                <line x1="50" y1="10" x2="50" y2="40" stroke="white" stroke-width="8" />
            </svg>
        }
    </button>
n>
</div>
<style>
/* ボタンスタイル CSS にしてもよい 例として razor ファイルに直接記載 */
button {
    background-color: transparent; /* 背景を透明に */
    border: none; /* 枠線をなくす */
    cursor: pointer; /* ホバー時にポインターを表示 */
    padding: 0; /* 余白をなくす */
}

/* 電源マークのスタイル */
svg {
    width: 50px; /* サイズを指定 */
    height: 50px;
    fill: none;
    stroke: black;
    stroke-width: 2;
}
</style>
@code {
#nullable disable
/// <summary>
/// コンポーネント名
/// </summary>
[Parameter]
public string TileComponent { get; set; } = "";
/// <summary>
/// コンポーネントタイプ
/// </summary>
[Parameter]
public int TileButtonType { get; set; } = 2;
/// <summary>
/// タグ名
/// </summary>
[Parameter]
public string TileTag { get; set; } = "";
/// <summary>
/// 表示文字列 (未使用)
/// </summary>
[Parameter]
public string TileText { get; set; } = "";
/// <summary>
/// タイルの配置 X 位置
/// </summary>

```

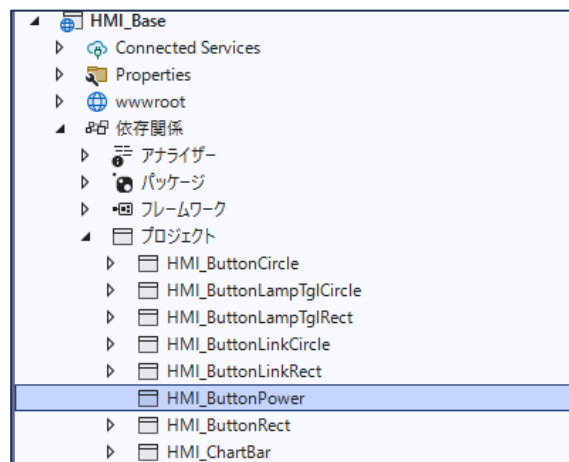
```

[Parameter]
public string TileLeft { get; set; } = "";
/// <summary>
/// タイルの配置 Y 位置
/// </summary>
[Parameter]
public string TileTop { get; set; } = "";
/// <summary>
/// タイルの ON/OFF
/// </summary>
[Parameter]
public string TileValue { get; set; } = "0";
/// <summary>
/// タイルの配置 Z 位置
/// </summary>
[Parameter]
public string TileZindex { get; set; } = "";
/// <summary>
/// クリックイベント
/// </summary>
[Parameter]
public EventCallback<(string TileComponent, string TileTag, string TileText, int TileButtonType, object
TileValue, int ValueType)> OnClick { get; set; }
/// <summary>
/// クリックイベント
/// </summary>
private async Task ButtonClick()
{
    var args = (TileComponent, TileTag, TileText, TileButtonType, TileValue, (int)egTagDataType.String);
    await OnClick.InvokeAsync(args);
}
#nullable restore
}

```

### 手順 3. タイルとコンテナ本体結合(HMI\_Base プロジェクトの編集)

- ① 依存関係に手順 2 で作成したプロジェクトを追加します。  
また「Components/\_Imports.razor」へ using 指定を追加します。



#### <\_Imports.razor>

```

@using System.Net.Http
@using System.Net.Http.Json
@using Microsoft.AspNetCore.Components.Forms
@using Microsoft.AspNetCore.Components.Routing
@using Microsoft.AspNetCore.Components.Web
@using static Microsoft.AspNetCore.Components.Web.RenderMode
@using Microsoft.AspNetCore.Components.Web.Virtualization
@using Microsoft.AspNetCore.Components.Authorization
@using Microsoft.JSInterop
@using HMI_Base
@using HMI_Base.Components
@using HMI_Base.Components.Tile

@* RT-edge *@
@* @using RTedge *@

```

```

@* タイル *@
@using HMI_LampCircle.Components.Pages
@using HMI_LampRect.Components.Pages
@using HMI_ButtonCircle.Components.Pages
@using HMI_ButtonRect.Components.Pages
@using HMI_LabelFix.Components.Pages
@using HMI_LabelTag.Components.Pages
@using HMI_LogViewTable.Components.Pages
@using HMI_ChartLine.Components.Pages
@using HMI_ChartBar.Components.Pages
@using HMI_Frame.Components.Pages
@using HMI_ButtonLampTglRect.Components.Pages
@using HMI_ButtonLinkRect.Components.Pages
@using HMI_ButtonLampTglCircle.Components.Pages
@using HMI_ButtonLinkCircle.Components.Pages
@using HMI_ButtonPower.Components.Pages

```

② タイル定義ファイル読み込み用クラスを作成します。

例では「HMI\_Base/Components/Xml」以下に、「HMI\_ButtonPower\_Xml.cs」を作成します。

- xml 読み込み処理を記載
- xml ファイル名は固定
- TileButtonType = 2 は固定値として記載

<HMI\_ButtonPower\_Xml.cs>

```

using System.Xml.Serialization;
using HMI_CommonLibrary;
using HMI_Base;

namespace HMI_Base
{
    /// <summary>
    /// 電源ボタン
    /// <remarks>
    /// 属性名は XmlAttribute で定義します。
    /// </remarks>
    public class HMI_ButtonPowerElem : HMI_TileElem
    {
        /// <summary>
        /// ComponentType
        /// </summary>
        [XmlAttribute("ButtonType")]
        public int ButtonType = 2;

        /// <summary>
        /// コンストラクタ
        /// </summary>
        public HMI_ButtonPowerElem()
        {
            // 初期設定
            Left = "0px";
            Top = "0px";
            ButtonType = 2;
        }
    }

    /// <summary>
    /// xml 情報
    /// </summary>
    public class HMI_ButtonPower_Xml
    {
        [XmlArrayItem("HMI_ButtonPower")]
        public List<HMI_ButtonPowerElem>? HMI_ButtonPowers = null;
    }

    /// <summary>
    /// xml のロード
    /// </summary>
    public partial class HMI_LoadXml
    {
        /// <summary>
        /// 電源ボタン xml ロード
        /// </summary>
        /// <param name="hmiButtonPowerXml"></param>
        static public void Load_HMI_ButtonPower_Xml(string folderName, ref HMI_ButtonPower_Xml hmiButtonPowerXml)
        {
            //ページフォルダ以下の xml を読み込む

```



```

hmiButtonPowerXml = LoadFromXml<HMI_ButtonPower_Xml>("${folderName}/HMI_ButtonPower.xml");
if (null != hmiButtonPowerXml)
{
    // 固定値設定
    if (null != hmiButtonPowerXml.HMI_ButtonPowers)
    {
        foreach (var Item in hmiButtonPowerXml.HMI_ButtonPowers)
        {
            Item.ButtonType = 2;//固定
        }
    }
}
}
}
}
}

```

- ③ タイル定義ファイル(xml)からロードした結果をページへ仮配置します。タイル定義ファイル(Xml)での指定が無い場合は配置されないよう名仕組みとなっています。
- 今回はボタンの為、「HMI\_Base/Components/Tile/DynamicTileButtonRenderer.razor」に追記します。新規でファイルを作成しても構いません。新規で追加した場合は、④で記載の個所に新規コードを追加します。以下、追記したコードです。

#### <DynamicTileButtonRenderer.razor>

```

@using HMI_Base
@using System.Collections.Generic

@code {
#nullable disable
    [Parameter]
    public required HMI_ButtonCircleGreen_Xml Tile_hmiButtonCircleGreenXml { get; set; }
    [Parameter]
    public required HMI_ButtonCircleRed_Xml Tile_hmiButtonCircleRedXml { get; set; }
    ... 省略 ...
    [Parameter]
    public required HMI_ButtonPower_Xml Tile_hmiButtonPowerXml { get; set; } //電源ボタン追加
    ... 省略 ...

@if (Tile_hmiButtonRectDarkblueOverXml != null && Tile_hmiButtonRectDarkblueOverXml.HMI_ButtonRectDarkblueOvers != null)
{
    @foreach (var Item in Tile_hmiButtonRectDarkblueOverXml.HMI_ButtonRectDarkblueOvers)
    {
        <Tile_ButtonRect TileComponent="HMI_ButtonRectDarkblueOver"
            TileFontSize="@Item.FontSize"
            TileLeft="@Item.Left"
            TileTop="@Item.Top"
            TileWidth="@Item.Width"
            TileHeight="@Item.Height"
            TileTag="@Item.Tag"
            TileText="@Item.Text"
            TileBackColor="@Item.BackColor"
            TileBackColorOver="@Item.BackColorOver"
            TileForeColor="@Item.ForeColor"
            TileValue="@Item.Value"
            TileZindex="@Item.Zindex"
            OnClick="ButtonClick" />
    }
}
@if (Tile_hmiButtonPowerXml != null && Tile_hmiButtonPowerXml.HMI_ButtonPowers != null)
{
    @foreach (var Item in Tile_hmiButtonPowerXml.HMI_ButtonPowers)
    {
        <Tile_ButtonPower TileComponent="HMI_ButtonPower"
            TileLeft="@Item.Left"
            TileTop="@Item.Top"
            TileTag="@Item.Tag"
            TileText="@Item.Text"
            TileValue="@Item.Value"
            TileZindex="@Item.Zindex"
            OnClick="ButtonClick" />
    }
}
}

```

- ④ ③の内容をメインページ(DynamicPage.raozr)から呼び出します。以下は、③の「DynamicTileButtonRenderer」への追加例です。新規で作成した場合は、同様に追加してください。

- ・ タイル配置処理追加
- ・ xml 読み込みクラスの変数追加
- ・ xml 読み込みクラス呼び出し処理追加
- ・ タイル値更新処理追加

<DynamicPage.razor>

```

...省略...
<DynamicTileButtonRenderer
    Tile_hmiButtonCircleGreenXml="hmiButtonCircleGreenXml"
    Tile_hmiButtonCircleRedXml="hmiButtonCircleRedXml"
    Tile_hmiButtonCircleRoyalblueXml="hmiButtonCircleRoyalblueXml"
    Tile_hmiButtonCircleOrangeXml="hmiButtonCircleOrangeXml"
    Tile_hmiButtonCircleDarkblueOverXml="hmiButtonCircleDarkblueOverXml"
    Tile_hmiButtonLampTglCircleGreenXml="hmiButtonLampTglCircleGreenXml"
    Tile_hmiButtonLampTglCircleRedXml="hmiButtonLampTglCircleRedXml"
    Tile_hmiButtonLampTglCircleBlueXml="hmiButtonLampTglCircleBlueXml"
    Tile_hmiButtonLampTglRectGreenXml="hmiButtonLampTglRectGreenXml"
    Tile_hmiButtonLampTglRectRedXml="hmiButtonLampTglRectRedXml"
    Tile_hmiButtonLampTglRectBlueXml="hmiButtonLampTglRectBlueXml"
    Tile_hmiButtonLinkCircleGreenXml="hmiButtonLinkCircleGreenXml"
    Tile_hmiButtonLinkCircleRedXml="hmiButtonLinkCircleRedXml"
    Tile_hmiButtonLinkCircleRoyalblueXml="hmiButtonLinkCircleRoyalblueXml"
    Tile_hmiButtonLinkCircleOrangeXml="hmiButtonLinkCircleOrangeXml"
    Tile_hmiButtonLinkCircleDarkblueOverXml="hmiButtonLinkCircleDarkblueOverXml"
    Tile_hmiButtonLinkRectGreenXml="hmiButtonLinkRectGreenXml"
    Tile_hmiButtonLinkRectRedXml="hmiButtonLinkRectRedXml"
    Tile_hmiButtonLinkRectRoyalblueXml="hmiButtonLinkRectRoyalblueXml"
    Tile_hmiButtonLinkRectOrangeXml="hmiButtonLinkRectOrangeXml"
    Tile_hmiButtonLinkRectDarkblueOverXml="hmiButtonLinkRectDarkblueOverXml"
    Tile_hmiButtonRectGreenXml="hmiButtonRectGreenXml"
    Tile_hmiButtonRectRedXml="hmiButtonRectRedXml"
    Tile_hmiButtonRectRoyalblueXml="hmiButtonRectRoyalblueXml"
    Tile_hmiButtonRectOrangeXml="hmiButtonRectOrangeXml"
    Tile_hmiButtonRectDarkblueOverXml="hmiButtonRectDarkblueOverXml"
    Tile_hmiButtonPowerXml="hmiButtonPowerXml"
    OnButtonClick="ButtonClick"
/>

...省略...
}
@code {
...省略...
// =====
// HMI_TextBoxSingle_Xml.cs 内に定義されているクラスを使用した変数
// =====
private HMI_TextBoxSingleWhiteBlue_Xml hmiTextBoxSingleWhiteBlueXml = null;
private HMI_TextBoxSingleWhiteGreen_Xml hmiTextBoxSingleWhiteGreenXml = null;
private HMI_TextBoxSingleWhiteOrange_Xml hmiTextBoxSingleWhiteOrangeXml = null;

// =====
// HMI_ButtonPower_Xml.cs 内に定義されているクラスを使用した変数
// =====
private HMI_ButtonPower_Xml hmiButtonPowerXml = null;

...省略...
//
// タイルの初期化(必要なものだけ)
//
private void InitTile()
{
    LoadLabelFix_Xml();
    LoadLabelTag_Xml();
    LoadButtonCircle_Xml();
    LoadButtonRect_Xml();
    LoadButtonLampRect_Xml();
    LoadButtonLampCircle_Xml();
    LoadButtonLinkRect_Xml();
    LoadButtonLinkCircle_Xml();
    LoadButtonPower_Xml(); // 電源ボタン タイル定義ファイル読み込み関数呼び出し
    LoadLampCircle_Xml();
    LoadLampRect_Xml();
    LoadLog_Xml();
    LoadChart_Xml();
    LoadFrame_Xml();
    LoadTextBox_Xml();
}

```

タイル配置処理

xml 読み込みクラス  
の変数追加

xml 読み込みクラス  
呼び出し処理追加

```

...省略...
/// <summary>
/// 電源ボタン xml ロード
/// </summary>
private void LoadButtonPower_Xml()
{
    // Xml ファイルの読み込み
    HMI_LoadXml.Load_HMI_ButtonPower_Xml(folderName, ref hmiButtonPowerXml);
}
/// <summary>
/// RT-edge I/F
/// </summary>
[Inject]
private EdgeIF egIF { get; set; }

/// <summary>
/// 入力用コンポーネントの更新
/// </summary>
private void UpdateInputComponent()
{
    //データを更新しない設定時は終了
    if(appSettings.Value.UpdateTile == false)
        return;
    // LabelTag を更新する
    UpdateLabelTag();
    // Lamp の状態を更新する
    UpdateLamp();
    // Log の状態を更新する
    UpdateLog();
    // Chart の状態を更新する
    UpdateChart();
    // IFrame の参照先を更新する
    UpdateIFrame();
    // 電源ボタンの値を更新する
    UpdatePowerValue();
}
...省略...
/// <summary>
/// 電源ボタンの値を更新
/// </summary>
private void UpdatePowerValue()
{
    string srVal = ""; //値読み込み変数
    if (null != hmiButtonPowerXml)
    {
        if (null != hmiButtonPowerXml.HMI_ButtonPowers)
        {
            foreach (var Item in hmiButtonPowerXml.HMI_ButtonPowers)
            {
                // タグの読み込み
                srVal = GetTag(Item.Tag, 0);
                // 1 ではない場合
                if (srVal != "1")
                {
                    // 該当のタグが Boolean 型で"True"を取得した場合は、ON とする
                    if (srVal == "True")
                    {
                        srVal = "1";
                    }
                    else
                    {
                        srVal = "0";
                    }
                }

                // 前回値との比較
                if (srVal == Item.Value)
                {
                    // 前回値と同じため処理終了
                    continue;
                }

                // 現在値の更新
                Item.Value = srVal;
            }
        }
    }

    /// <summary>
    /// 指定した実数のフォーマットを取得する
    /// </summary>
    ...省略...

```

タイル値更新処理  
追加

#### 手順 4. タイル定義ファイル(xml)作成

「[手順 3. タイルとコンテナ本体結合](#)」の XML 読み込みクラスで定義した内容の通り、タイトル定義ファイル(XML)を作成する。例ではタグ名「Command\_Power」を指定。

```
<?xml version="1.0" encoding="utf-8"?>
<HMI_ButtonPower_Xml xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <HMI_ButtonPowers>
    <HMI_ButtonPower Tag="Command_Power" Text="" Top="50px" Left="200px" />
  </HMI_ButtonPowers>
</HMI_ButtonPower_Xml>
```

#### 手順 5. 配置・表示

作成したタイトル定義ファイル(XML)を配置する。「HMI\_ButtonPower.xml」ファイルを読み込むように記載している為、同名のファイルを配置する。以下はトップページ上に配置する例を記載。

(ア) Visual Studio デバッグ実行時

<HMI\_Project 配置フォルダ>%0010\_HMI\_Base%bin%Debug%net8.0%wwwroot%\_xmls%root% HMI\_ButtonPower.xml

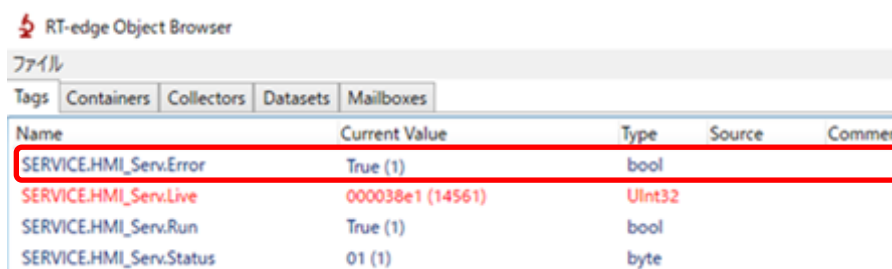
(イ) IIS 上で実行時

<IIS 配置フォルダ>%wwwroot%\_xmls%root%HMI\_ButtonPower.xml

# 12. 付録

## 12.1. トラブルシューティング

サービスインジケータの.Error が true になっています。



Name	Current Value	Type	Source	Comment
SERVICE.HMI_Serv.Error	True (1)	bool		
SERVICE.HMI_Serv.Live	000038e1 (14561)	UInt32		
SERVICE.HMI_Serv.Run	True (1)	bool		
SERVICE.HMI_Serv.Status	01 (1)	byte		

図 14. Error インジケータが true

「.Error」タグが True になる場合、内部的に問題がある状態となります。  
一度すべてのコンテナを終了し、再度起動してください。

サービスインジケータの.Run が false になっています。

### ■ 原因：サービス開始要求を受け付けられていません。

サービス開始要求が無いが、初期化処理が終わらずサービス開始要求を受け付けられていません。

#### ➤ 対応：サービス開始要求を行います。

アプリケーションから「EM\_SERVICE\_RUN」メッセージを送る、または  
HMI\_Server.xml にて「.AutoRun」プロパティの Value を「1」とします。

ブラウザにタイルは表示されるが、タグの値が反映されません。

### ■ 原因：HMI\_Server コンテナが起動していません。

HMI\_Server コンテナを起動していない場合は画面上にタグの値は表示されません。

#### ➤ 対応：サービス開始要求を行います。

HMI\_Server コンテナを起動してください。

### ■ 原因：登録されていないタグが指定されています。

HMI\_Server は起動している状態で値が表示されない場合は、登録されていないタグが指定されている可能性があります。

#### ➤ 対応：タグの定義を行います。

RT-edge 側でのタグの定義、タイルで指定しているタグの定義を確認してください。



更新履歴

版	日付	更新説明
1	2024.11	初回版
2	2025.8	RTCD の名称を「RT-edge Object」に変更

INDUSTRIAL REALTIME EDGE COMPUTERS

HMI コンテナ ユーザーズマニュアル

発行元:株式会社マイクロネット  
TEL: +81(0)299-90-1733

- ・ 本書の内容、及び付属のソフトウェアの全部または一部を無断で転載することは禁止しております。
- ・ 本製品の内容については、将来予告なしに変更することがあります。
- ・ 本製品の内容について万一ご不審な点や記載もれなどお気づきの点がございましたら、お手数ですが、当社までご連絡ください。
- ・ Windows XP、Windows 7、Windows 8、Windows 10 等、Windows は、米国 Microsoft Corporation における登録商標です。
- ・ Visual Studio、Visual C++等は、米国、およびその他の国における Microsoft Corporation の登録商標です。
- ・ INtime は米国 TenAsys における登録商標です。
- ・ その他、記載されている会社名、製品名は、各社の商標又は登録商標です