

第1章 「楽々ウェブクライアント」概要

1.1 「楽々ウェブクライアント」の特長

楽々ウェブクライアントは、ソース管理ツール「SMSYS」のサブセットです。
 ソース管理ツール「SMSYS」は、MAGICのプロジェクトファイルにアクセスし、モデル、データ、プログラム等のデータをアイテム別にシュレディングし、RDBMSに格納します。ソースビューアとしての機能を有しながら、開発作業を支援する様々な機能を提供しています。
 楽々ウェブクライアントは、MAGIC xpa Ver4以降でサポートされたWebClientに特化した開発支援機能をサポートします。

おおよそ、次のような特長を持っています。

MAGICのソース管理ツール「SMSYS」をバックエンドに、リポジトリ情報をベースとしたAngularソースの変換処理をサポート

MAGICが出力するAngular ソースは、そのままではデザイン情報を有していないため、何らかのソース編集が必要になります。
 しかし、ソースを手作業により修正するのは、大変な作業になります。MAGICのフォームとフォームに配置したコントロールを視覚的に捉えながらダイナミックに編集・変換する機能を提供します。
 デザイン情報はデータとして残されるので、不測のエラーが起きてもプロジェクト全体 / 個別の再構築が容易です。

MAGIC xpaが生成したAngularソースに直接アクセスし、Angularソースの改変を強力にサポート

手作業の開発では、ソースファイルの管理を怠ると、ファイルの上書き等により、デザインした編集内容を失ってしまいかねません。
 当アプリケーションの環境下では、TSファイル内のテンプレート指定を自動的に書き変えます。また、テンプレートの更新検知、TSファイルのバックアップとリストア、CSS作成等の自動化により、面倒なソースファイルの管理作業から解放されます。

Bootstrapを利用したWebデザインをサポートし、環境構築やBootstrapのクラスに対応したソースを自動生成

BootstrapはHTML / CSS / JavaScriptから構成される最も有名なWEBフレームワークです。
 コントロールに対して、GUI的な操作でBootstrapのクラスを割り当てる属性編集機能を提供します。
 フォーム全体 (もしくはフォーム中の幾つかのエリア) に対しても予め定義した変換ルールを適用することでテンプレートの一括変換が可能です。
 これらの機能により、ソースを直接編集することなくデザイン作業を行うことが可能です。

1.2 「楽々ウェブクライアント」の目的

1.2.1 MAGIC xpa WebClientによる開発の現状

WebClientは、Angularを搭載したSPA型のWebアプリケーションを構築可能にする MAGIC xpa Ver4になって実装された画期的な機能です。
 画面デザインとロジックを分離できるメリットがある反面、MAGICとAngularの二重のソースコードの管理が必要になります。
 このソースの二重管理をどのように解消するかということが特に開発の初期の局面では大きな課題となります。
 これはAngularプロジェクトを作成するための処理「Webアプリケーションの作成」が次のようなMAGICの仕様に起因しています。

- ・基本的にAngularソースの生成が「MAGIC → Angular」の一方向である
- ・既に作成してあるAngularソース (*.html, *.ts) を (修正の有無に関わらず) 無条件に上書きしてしまう
- ・Angularソースがフォーム単位であるのに対して、プログラム単位でしかソース作成ができない。例えば、変更したサブタスクのみ更新しようと思ってもそれができないために、不用意に関係のないソースまで上書きしてしまわないような、細心の注意が必要になってしまう
- ・MAGICのフォーム名、フォルダ名等を変更した時など、Angularソースを手作業で修正しなければならないことがある
 (例: 前の名前が残ったソース部分を除去する等)

この結果、MAGIC側のロジックの完成度を高めたのちにAngularでの開発に移行するなど、設計変更を極力避ける必要に迫られてしまうことになります。

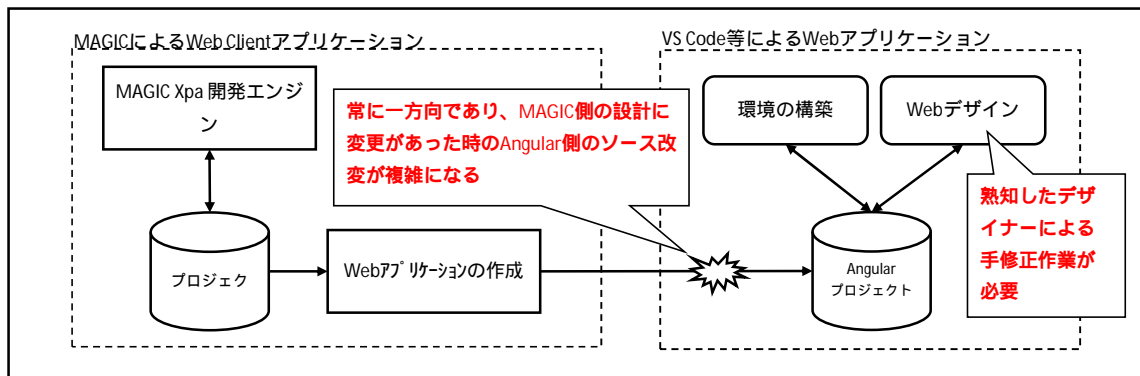


図1.2.1-1 一般的なWeb Client アプリケーション開発の流れ

1.2.2 問題の解決

若干漠然とはしていますが、下記のようなシステムの作成と利用により、プロトタイプ作成に必要なレベルでのコーディングの自動化を目指します。

- Visual Studio Code等によるソースの手修正を必要としない、環境設定、ソース改変の自動化
MAGICのソースとAngularのソースを統合的に管理し、並行的に動作する開発環境を提供
- デザイン情報のデータ化、ひな形のテンプレート化
GUI画面によるデザイン修正と、ルール / デザイン情報のデータ化により、何度でも同じように更新を再現できる仕組みの構築
- リポジトリ情報の積極的な利用
編集画面はMAGICの開発画面に近いものにする。また、リポジトリ情報を利用した編集支援機能を提供
- Bootstrapの積極的な利用
画面のレイアウトや個々のオブジェクトのデザインをBootstrapのクラスを使うことによりCSSの知識等が無くても簡単に利用できる

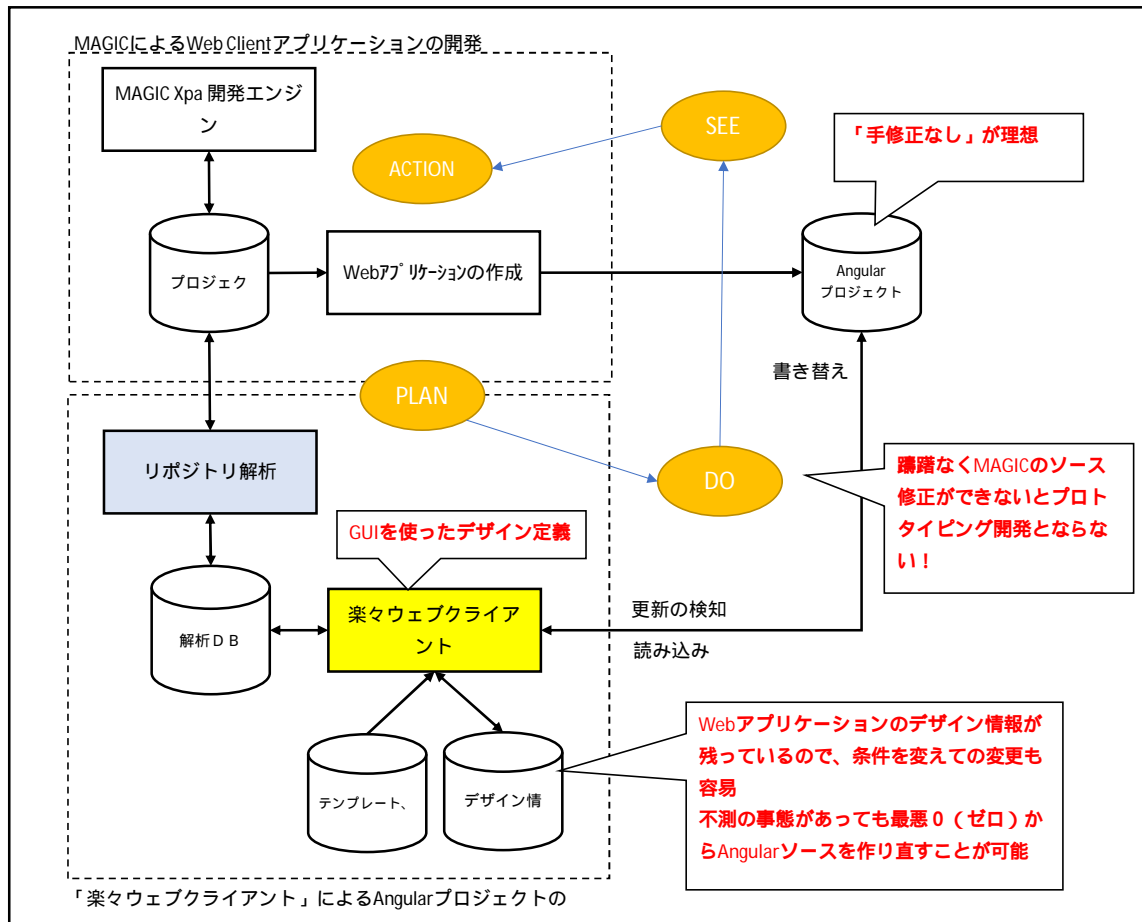


図1.2.2-1 「楽々ウェブクライアント」を利用したWeb Clientアプリケーション開発

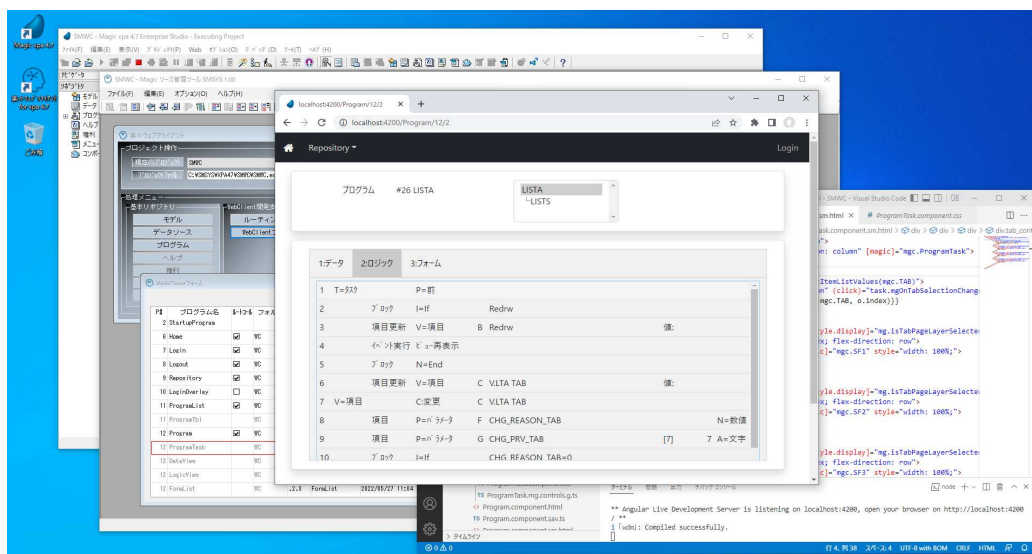


図1.2.2-2 「楽々ウェブクライアント」はMAGIC + VSCode + ブラウザの開発環境の中で並行的に動作

1.3 「楽々ウェブクライアント」がカバーする範囲

1.3.1 Angular環境構築

Bootstrapのインストールを始め、Bootstrapのバージョンに対して設定した情報を元に、ウィザードを使用して環境の構築が可能です。
npmモジュールのインストール、angular.jsonの設定、src/index.htmlの設定、src/styles.cssの設定等を自動化します。



図1.3.1-1 Bootstrap設定ウィザードによる環境自動設定

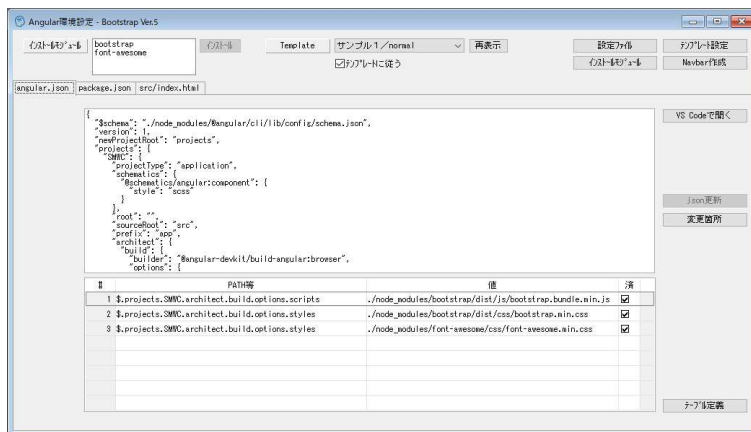


図1.3.1-2 Angular環境設定によるウィザードデータの設定

1.3.2 ナビゲーションバーの作成

ウィザードによりBootstrapのNavbar自動作成が可能です。
予め設定されたテンプレートを選択して行うことが可能です。
MAGICのルーティング情報(スタートアッププログラムで設定した「ルート」イベントのプログラム)によるメニューの定義が可能です。

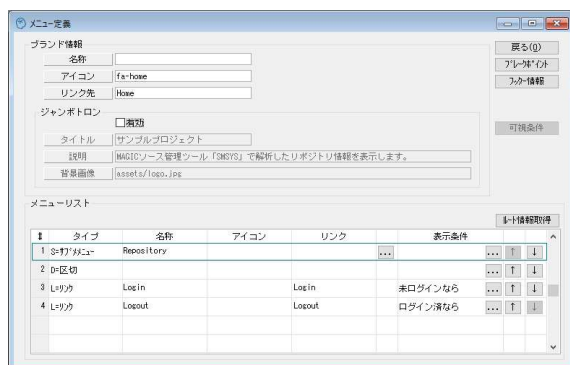


図1.3.2-1 Bootstrap設定ウィザードのメニュー設定

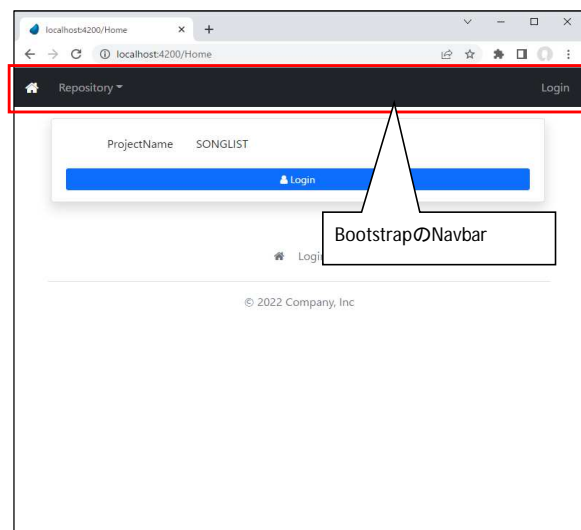


図1.3.2-1 作成されたナビゲーションバーの例

1.3.3 リポジトリ管理

1) リポジトリ情報のデータ化

ソース管理ツール「SMSYS」はプロジェクトのリポジトリ情報（モデル、データ、プログラムやその関連）をシュレディングして複数のテーブルに格納します。MAGIC xpa 4.xのWebClientへ対応するために拡張されています。

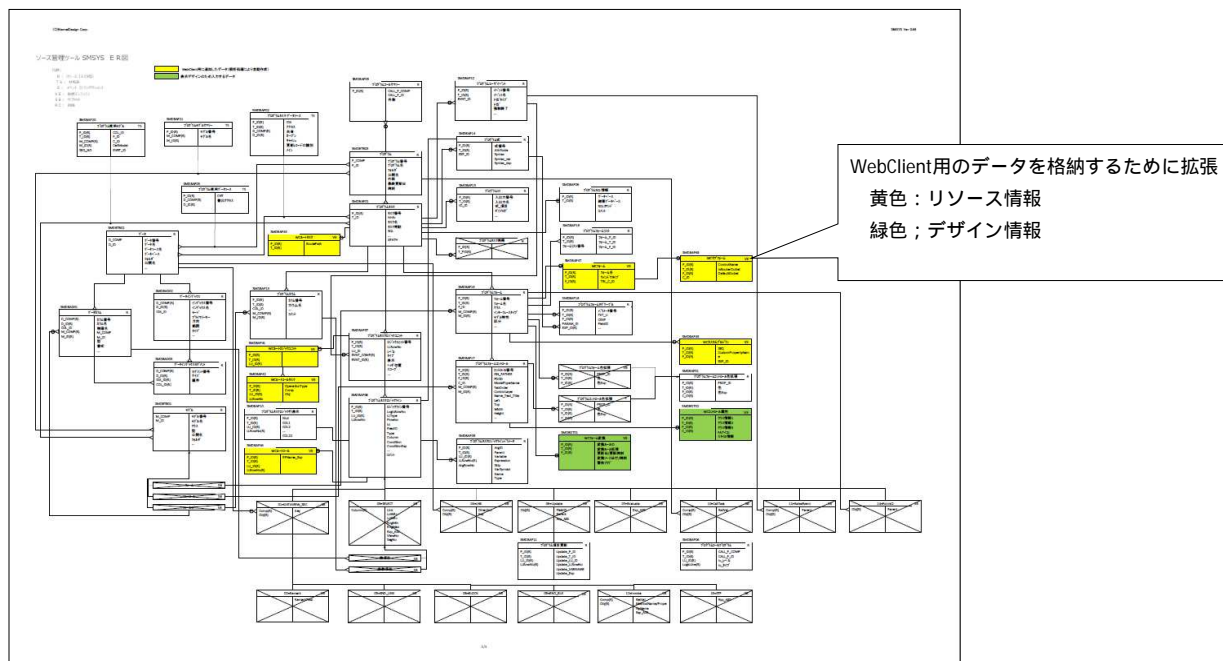


図1.3.3-1 SMSYSのERダイアグラム

2) ルーティング情報

ルーティングに関する情報をひとつのプログラムで確認することが容易になっています。

スタートアッププログラム/ルート定義
プログラム名: StartupProgram, P#: 2, フォルダ名: StartupProgram, 実行: 1, 更新時刻: 2022/05/27 11:02

行#	コード名	番号	コール先名	ルート名	出力先	式	条件	フォルダ	最終更新時刻
4	Home	6	Home	Home	WC			WC	2022/05/27 11:02
5	Login	7	Login	Login	WC			WC	2022/05/27 11:03
6	Logout	8	Logout	Logout	WC	1	UserHomeOut	WC	2022/05/27 11:03
7	Repository	9	Repository	Repository	WC			WC	2022/05/27 11:03
8	ProgramList	11	ProgramList	ProgramList	WC			WC	2022/05/27 11:03
9	Program	12	Program	Program	WC			WC	2022/05/27 11:04

全フォーム(プログラム)

行#	プログラム名	フォルダ	サブフォルダ	F#	フォルダ名	出力先	WebClient	処理
12	Program		2	Program	Program	WC	WC	変換
12	Program	.1	ProgramTask	3	ProgramTask	WC	WC	変換
12	Program	.1.1	DataView	4	DataView	WC	WC	変換
12	Program	.1.2	LogicView	4	LogicView	WC	WC	変換

図1.3.3-2 ルーティング情報画面

3) 全フォームの一覧化

プロジェクト内の全てのフォームをプログラム番号順に確認することが可能です。

行#	プログラム名	フォルダ	サブフォルダ	タスク名	最終更新時刻	F#	フォーム名	OP	CV	出力先	WebClient	処理
2	StartupProgram				2022/05/27 11:02	2	StartupProgram	WC	WC	WC	WC	変換
6	Home	WC			2022/05/27 11:02	2	Home	WC	WC	WC	WC	変換
7	Login	WC			2022/05/27 11:03	2	Login	WC	WC	WC	WC	変換
8	Logout	WC			2022/05/27 11:03	2	Logout	WC	WC	WC	WC	変換
9	Repository	WC			2022/05/27 11:03	2	Repository	WC	WC	WC	WC	変換
10	LoginOverlay	WC			2022/05/27 11:03	2	LoginOverlay	WC	WC	WC	WC	変換
11	ProgramList	WC			2022/05/27 11:03	2	ProgramList	WC	WC	WC	WC	変換
11	ProgramList	WC	.1	ProgramTbl	2022/05/11 16:20	3	ProgramTbl	WC	WC	WC	WC	変換
12	Program	WC			2022/05/27 11:04	2	Program	WC	WC	WC	WC	変換
12	Program	WC	.1	ProgramTask	2022/05/27 11:03	3	ProgramTask	WC	WC	WC	WC	変換
12	Program	WC	.1.1	DataView	2022/05/27 11:03	4	DataView	WC	WC	WC	WC	変換
12	Program	WC	.1.2	LogicView	2022/05/27 11:03	4	LogicView	WC	WC	WC	WC	変換
12	Program	WC	.1.3	FormList	2022/05/27 11:04	4	FormList	WC	WC	WC	WC	変換
13	DataList	WC			2022/06/08 16:28	2	ProgramList	WC	WC	WC	WC	変換
13	DataList	WC	.1	DataTbl	2022/06/08 16:28	3	DataTbl	WC	WC	WC	WC	変換

図1.3.3-3 WebClientフォーム画面

4) エラーチェック機能

ソースの解析を行う時点で下記のチェックを行います。

- ・コントロール名に漢字等の使えない文字が使用されていない
- ・ルーティング設定されたプログラムのパラメータとなる項目は「ルートに出力」が設定されている

エラーが存在する場合は、画面上でそれを直ちに確認することができます。

コントロール名に漢字等の使えない文字が使用された場合

「WebClientフォーム」プログラムでは、エラーのあるフォームのフォーム名の表示色が変わります。

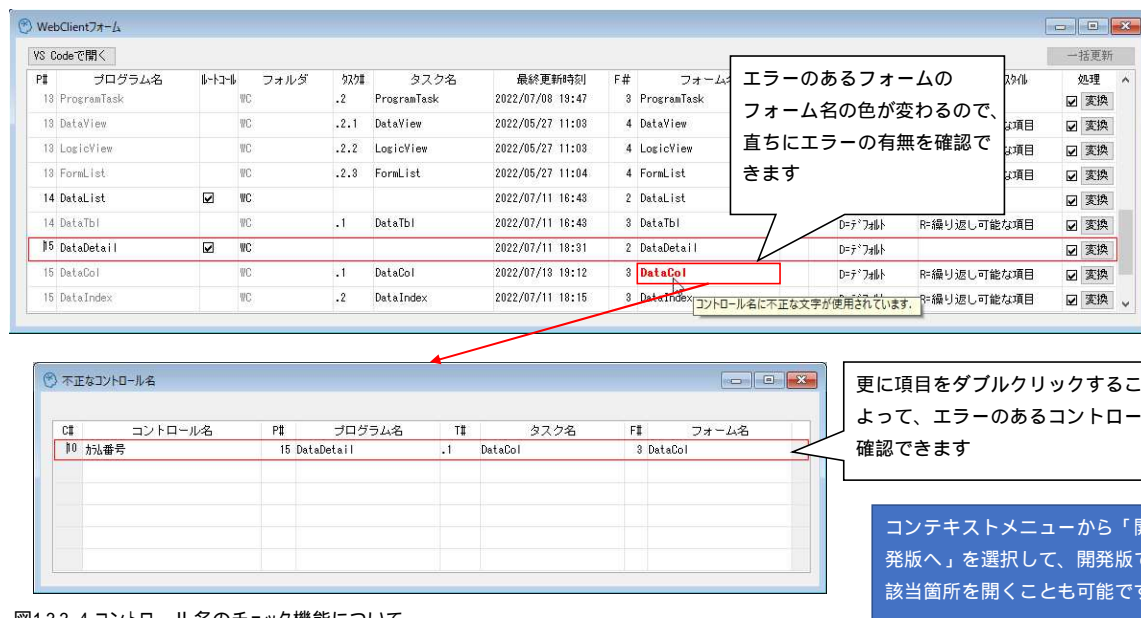


図1.3.3-4 コントロール名のチェック機能について

ルーティング設定されたプログラムのパラメータとなる項目に「ルートに出力」が設定されていない

「ルーティング情報」プログラムでは、ルート定義明細のエラーのあるコール先名の「パラメータ」の表示色が変わります。

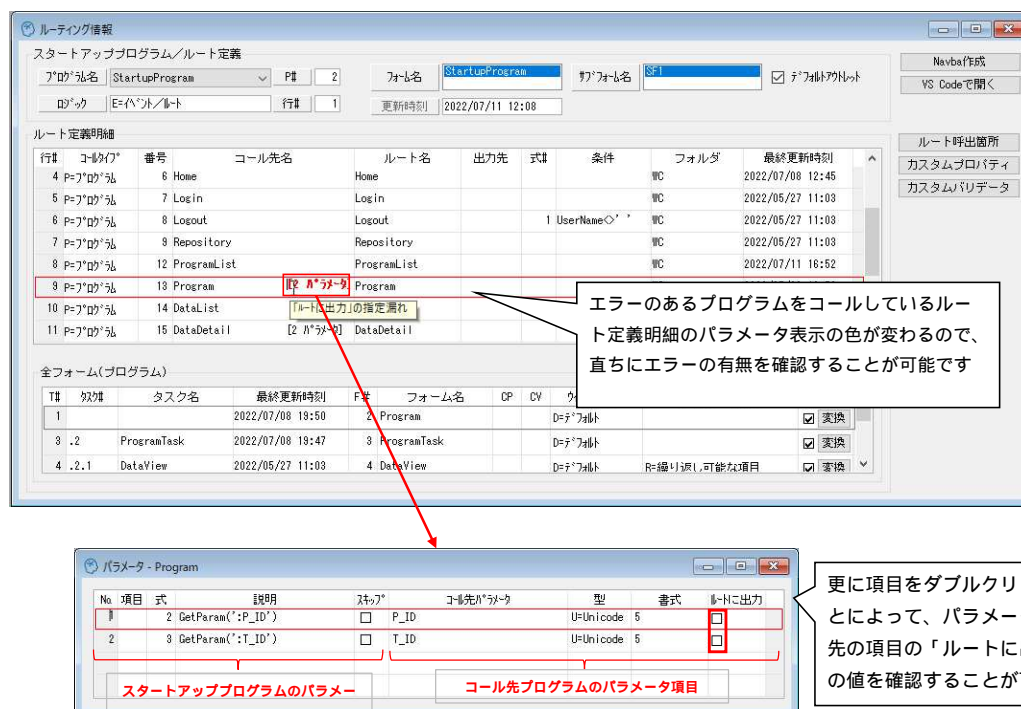


図1.3.3-4 パラメータの属性「ルートに出力」のチェック機能について

1.3.4 テンプレートの変換

テンプレートの変換は「楽々ウェブクライアント」のメインとなる機能です。
次のような特長があります。

変換ルールを利用してダイナミックに変換が可能です

変換ルールとは、HTMLに対する要素や属性の追加 / 変更 / 削除を予め定義した一連のセットで、ルールを適用することにより一括的に変換を行うもので、テーブルの有無やテーブルの種類別に定義しており、自動的に適用可能な変換ルールを選択することが可能です。
テーブル(繰り返し可能な項目)であればcardクラス、tableクラスを使用したテンプレートが準備しており、ルールやクラス情報の属性の変更も簡単に行えます。
複雑なフォームもエリア別に変換ルールを適用させることが可能です。(MAGICで作成されたテンプレートから自動判別します。
独自の変換ルールを作成したり編集が可能です。(正規版のみ対応)

コントロールに対するクラス属性の割り当てがGUI画面から行えます

プッシュボタン / エディット / ラベル / ラジオボタン / コンボボックス等のコントロール種別毎にBootstrapの指定可能なクラス属性を定義することが可能です。
Bootstrapのグリッドシステムを使用して配置が行えます。
コントロールの表示 / 非表示をコントロールすることが可能です。
カスタムプロパティの値を使用したHTMLの生成が可能です。(ラベル等の特定のコントロール)
Bootstrapのクラス属性を直接指定することが可能です。(正規版のみ対応)

CSS編集補助機能

テーブルであれば、高さや水平スクロールバーの有無をダイアログ画面で指定が可能です。
CSSの直接編集が可能です。編集結果はデータベースに保存しますので、復元が可能です。(正規版のみ対応)
CSS使用時のTSファイルの更新は自動で行います。

TSファイルの編集補助機能

バックアップ / リストアの自動化により、不用意な書き換えにも対応が可能です。
オーバーレイタイプのモーダルウィンドウのサイズやタイトルを指定することが可能です。
VS Codeで直接編集した場合は、ダイアログを使用してバックアップが可能です。

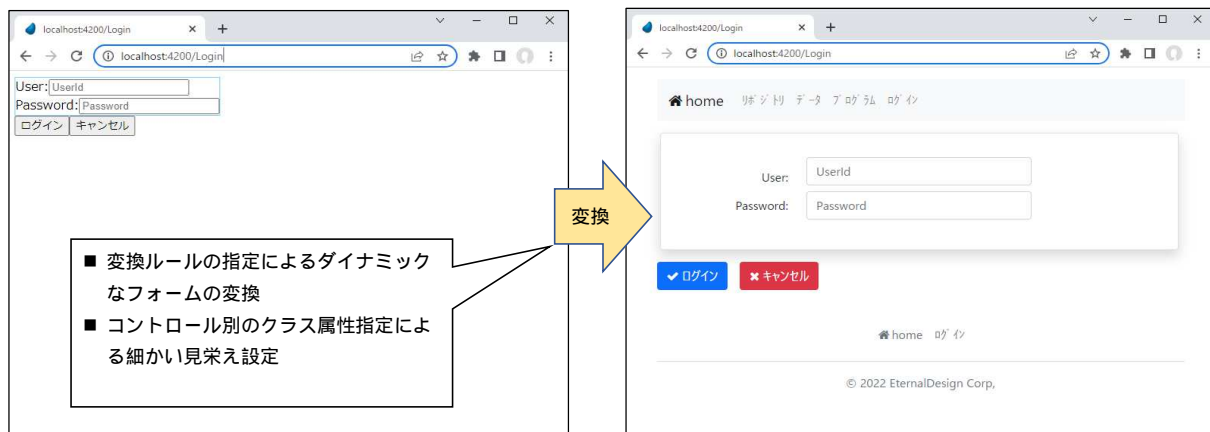


図1.3.4-1 「テンプレートの変換」によるHTMLデザインの書き換え処理例

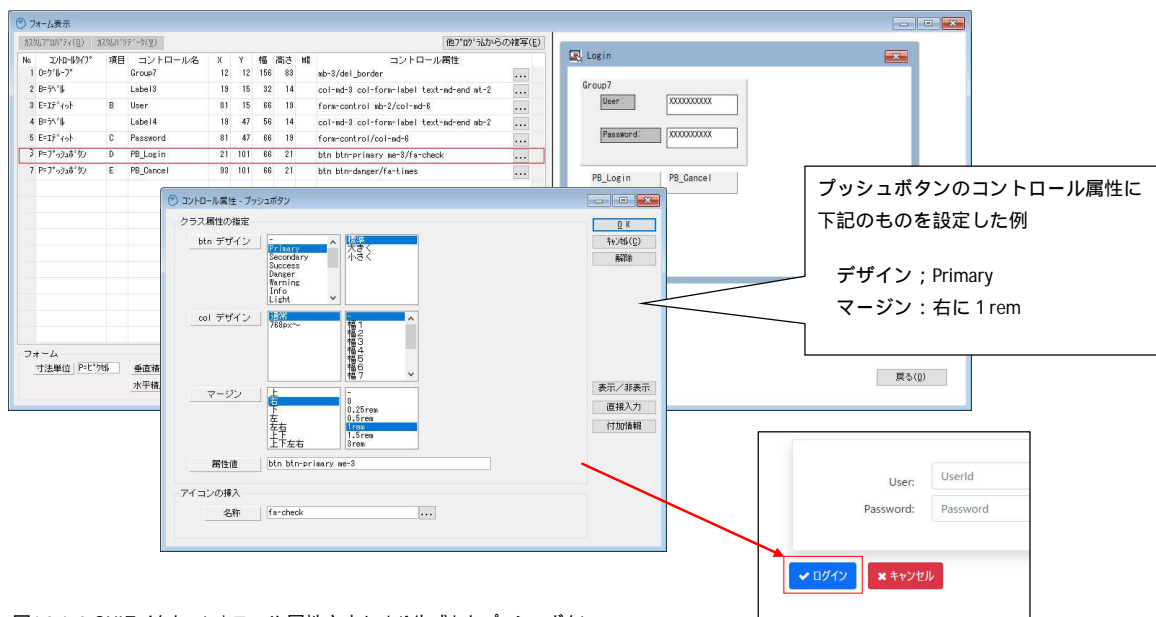


図1.3.4-2 GUIライクなコントロール属性入力により生成したプッシュボタン

1.4 動作条件

「楽々ウェブクライアント」の動作条件は下記の通りです。

動作環境	MAGIC xpa ver4.xの開発環境 (WebClientが動作している必要があります)
RDBMS	Microsoft SQL Server Version 2016以上 2016より古いバージョンは利用できません SQLiteはサポート対象外です

1.5 評価版と正規版の違いについて

インストール直後は「評価版」として動作します。
ソース管理ツール「SMSYS」の基本機能に関しては継続してご利用頂くことが可能ですが、「楽々ウェブクライアント」に関しては評価期限が設定されます。
継続してご利用頂くためには評価期限内にライセンスをご購入下さい。
評価版は、下記の機能制限を設けています。

評価版の制限と有効期限について

評価版では実行できない機能	<p>テンプレート変換の下記の機能</p> <p>変換ルール編集機能 (「変換ルールTBL保守」ボタン)</p> <p>CSS直接編集機能 (「CSS直接記述」ボタン)</p> <p>「コントロール属性」編集時のクラス属性の「直接入力」</p> <p>クラス属性を直接入力する機能</p> <p>Angular環境構築のための設定プログラム</p> <p>自動インストールするためのnpmモジュールや関連ファイルへの自動組み込み設定 (ライセンスが有効な場合に、プルダウンにメニューを表示します)</p>
評価期限について	<p>インストール後、3か月</p> <p>有効期限の3週間前になると下記のメッセージを表示します。</p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>評価期限 (YYYY/MM/DD) が近付いています。</p> <p>継続してご利用頂くためにはライセンスをご購入下さい。</p> <p>お問い合わせ Email: support@eternaldesign.jp</p> </div> <p>有効期限が切れた場合はプログラム「テンプレートの変換」プログラムの実行ができなくなります。</p>
正規ライセンスの適用	送付されたライセンスファイルを「ライセンス登録」プログラムで読み込ませてください

第2章 「楽々ウェブクライアント」の導入

2.1 アーカイブの展開と開発版でのインストール

2.1.1 アーカイブの解凍

ファイル「SMSYS.X###.zip」を任意のフォルダに展開します。

フォルダイメージ

SMSYS.X###	
├ SM.SETUP	インストール用プロジェクト「SM.SETUP」フォルダ
│ └ SM.SETUP.edp	「SM.SETUP」プロジェクトファイル
│ └ Data	SMSYSインストールデータ
│ │ └ DEVELOP	SMSYSインストールデータ(開発用データ)
│ │ └ DLL	SMSYSインストールデータ(DLLファイル)
│ │ └ ECF4	SMSYSインストールデータ(xpa 4.x 用キャビネットファイル)
│ │ └ ECF32	SMSYSインストールデータ(xpa 3.2 用キャビネットファイル)
│ │ └ ECF33	SMSYSインストールデータ(xpa 3.3 用キャビネットファイル)
│ └ Exports	「SM.SETUP」Exportsフォルダ(空)
└ Source	「SM.SETUP」ソースフォルダ

2.1.2 開発版の起動

動作環境の実行モードが「B=バックグラウンド」に設定されている場合はプログラムが実行できません。
予め、「O=オンライン」に変更して下さい。

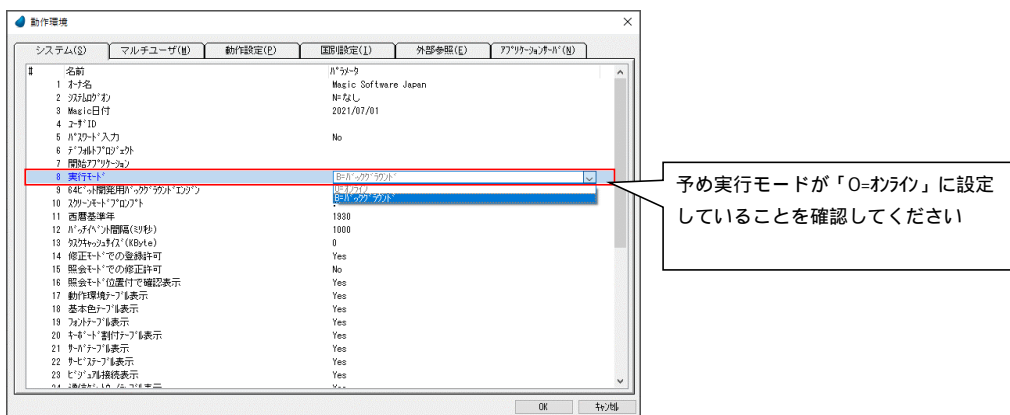


図2.1.2-1 MAGIC xpa 開発版の「動作環境」画面

起動用のショートカットの「リンク先」にオプションを加えて起動してもオンラインモードで起動することができます。

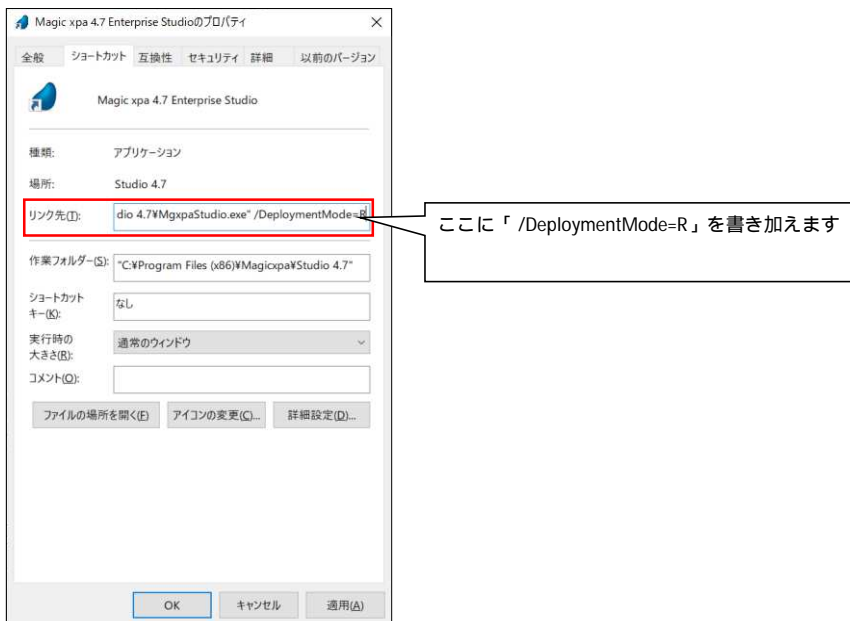
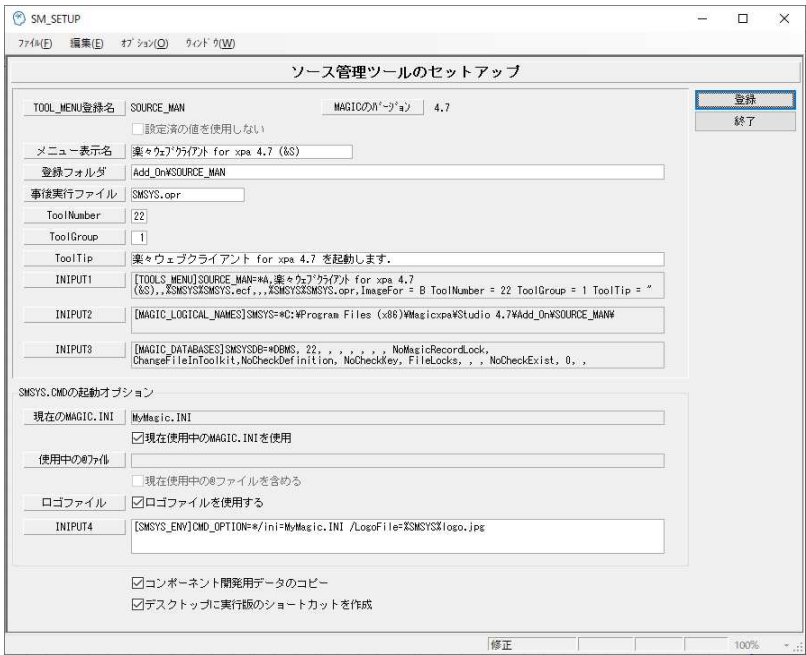


図2.1.2-2 ショートカットを変更してオンラインモードで起動させる

2.1.3 プロジェクトの実行

で展開したフォルダのプロジェクトファイルを選択し、「SM.SETUP」を開きます。
「プロジェクトの実行」を起動します (もしくはプログラムリポジトリから#2「SETUP」を実行します)



MAGIC.INIは指定しない環境での運用をお勧めします。
何らかの理由でMAGICが再起動したとき、「/INI=」オプションは無視されてしまうためです。

図2.1.2-3 セットアッププログラム起動画面

各項目の意味は下記の通りです。

セットアッププログラムの項目		
項 目	説 明	
設定済みの値を使用しない	新規設定時は無効です。既にユーザ定義開発メニューの設定がある場合、このチェックを有効にすることで初期値をデフォルト値に戻すことが可能です。	
メニュー表示名	ユーザ定義開発メニューに表示する名称を設定します。	
登録フォルダ	インストール先のフォルダを指定します。 初期値は、MAGIC xpaのシステムフォルダ直下の「Add_On¥SOURCE_MAN」です。 「F5:ズーム」キーで任意のフォルダを選択することが可能です。	
事後実行ファイル	ユーザ定義開発メニューのオプション	
ToolNumber	ツールバーに表示するアイコンの番号	
ToolGroup	ツールバーに表示するアイコンのグループ番号	
ToolTip	ツールバーに表示するアイコンのツールチップ	
現在のMAGIC.INI	起動時のコマンドにINIの指定がある場合は「現在のMAGIC.INI」に内容を表示します。 「現在の使用中のMAGIC.INIを使用」がチェック状態になり、内容が「INPUT4」にセットされます。 INI指定時はチェックを外すと矛盾が生じるので外さないで下さい。	
使用中の@ファイル	起動時のコマンドに@ファイルの指定がある場合は「使用中の@ファイル」に内容を表示します。 「現在の使用中の@ファイルを含める」にチェックすると内容が「INPUT4」にセットされます。	
ロゴファイル	起動時にロゴファイルを表示させる場合はチェックします。 ロゴファイルのファイル名は「INPUT4」の「/LogoFile=ファイル名」で指定します。	
コンポーネント開発用データのコピー	SMSYSをコンポーネントとして利用する場合にチェックします。 プロジェクトファイルとECIファイルをインストールします。 また、SMSYSを利用したWebClientのサンプルプロジェクトをインストールします。 (後の説明で使いますので、インストールして下さい)	
デスクトップに実行版のショートカットを作成	チェックした時は実行版起動用のショートカットを作成します。	

内容を確認し、「登録」ボタンを押してプログラムを終了します。

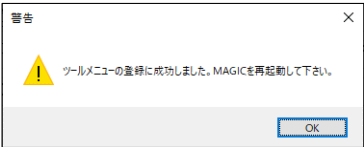


図2.1.2-4 再起動確認のダイアログ

MAGICを再起動します。
プルダウンメニュー「ツール(T)」にメニューが表示されることを確認して下さい。
同様に、ツールバーを表示している場合は、アイコンが追加されます。



図2.1.2-5 インストール後のプルダウンメニュー (ツール)



図2.1.2-6 インストール後のツールバー

< 補足 >

「デスクトップに実行版のショートカットを作成」オプションを指定時、ショートカットのアイコンが正しく表示されない場合は次の手順で修正してください。

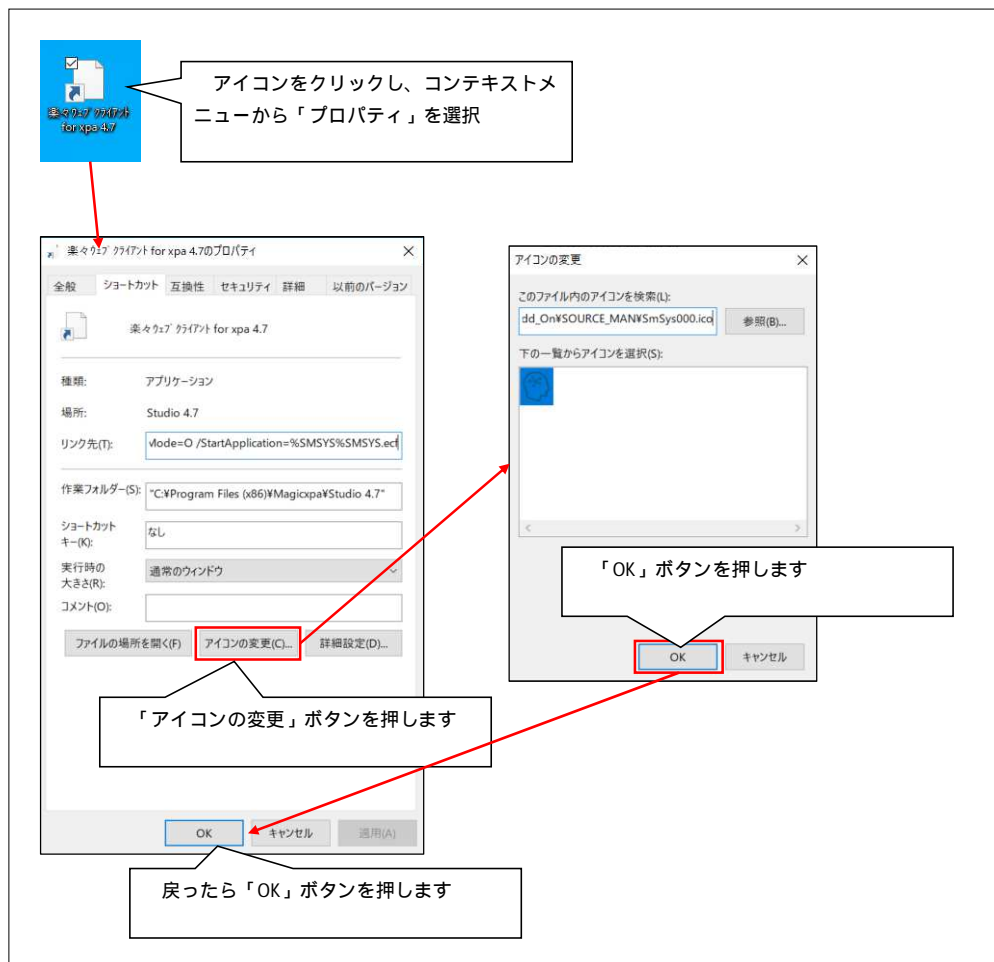


図2.1.2-7 ショートカットのアイコン修復手順

2.2 サンプルプロジェクトによる環境構築

サンプルプロジェクトを使用し、環境を構築するまでの手順を示します。

2.2.0 MAGIC xpa studioによるWebclient環境の構築

前提として、WebClient環境の構築が条件です。
予め簡単なプログラム等で WebClient が動作することを確認しておきます。

インストール

Node.jsをインストール
Angular cliをバージョン指定でインストール(npm install -g @angular/cli@8.3.25)
VisualStudio Codeをインストール
Chromブラウザをインストール
Magic xpa4.Xをインストール

MAGICの環境設定

アプリケーションサーバーとして動作
実行モードを「B=バックグラウンド」に設定

2.2.1 サンプルプロジェクトのコピー

インストールしたフォルダ「PROJ」の下にある「SMWC」を適当なフォルダにコピーします。
以下は「C:\DEMO」にコピーしたという前提で説明を進めます。

2.2.2 プロジェクトを開発版で開く

「Webアプリケーションの設定」を確認します。

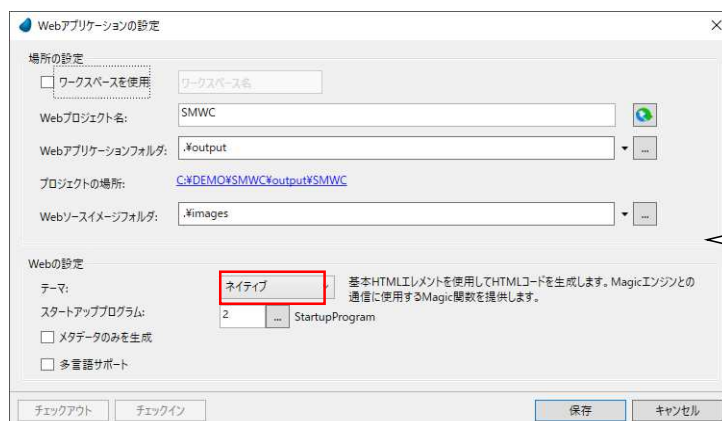


図2.2.2-1 Webアプリケーションの設定画面

テーマは「ネイティブ」に設定されていることを確認します

評価版で前提としているテーマは「ネイティブ」なので、もしサンプルプロジェクト以外のプロジェクトで利用するためにはテーマを「ネイティブ」に変更する必要があります。

2.2.3 Webアプリケーションの作成

「Webアプリケーションの作成」を起動します。
ルーティングマップを生成をチェックします。
「プログラム」を「All - 全て」に設定し、「生成」ボタンを押します。

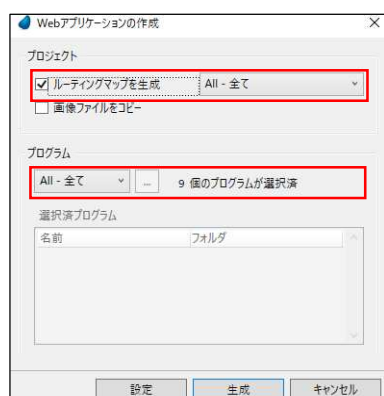


図2.2.3-1 Webアプリケーションの作成ダイアログ

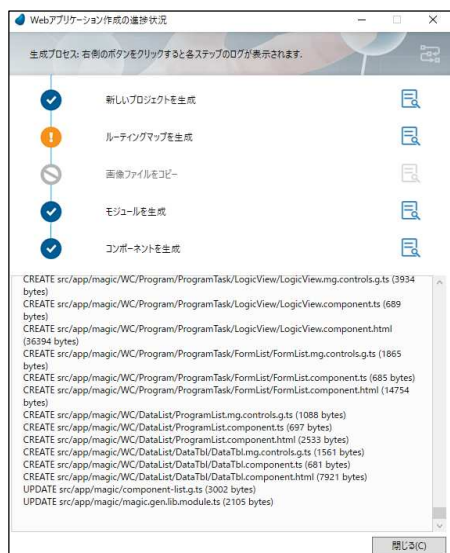


図2.2.3-2 Webアプリケーションの作成の進捗状況

2.2.4 プロジェクトの実行による動作確認

「プロジェクトの実行」から実行エンジンを起動しておきます。
 「Visual Studio Codeで開く」を実行し、作成したAngularプロジェクトを開きます。
 ターミナルからサーバーを起動します。

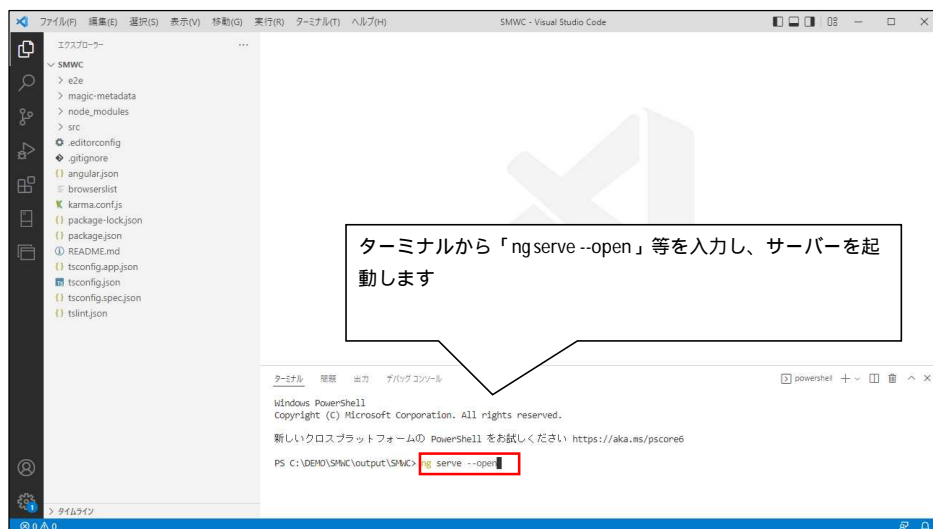


図2.2.4-1 Visual Studio Codeを起動し、ターミナルからサーバーを起動

ブラウザでプログラムを起動し表示されることを確認します。

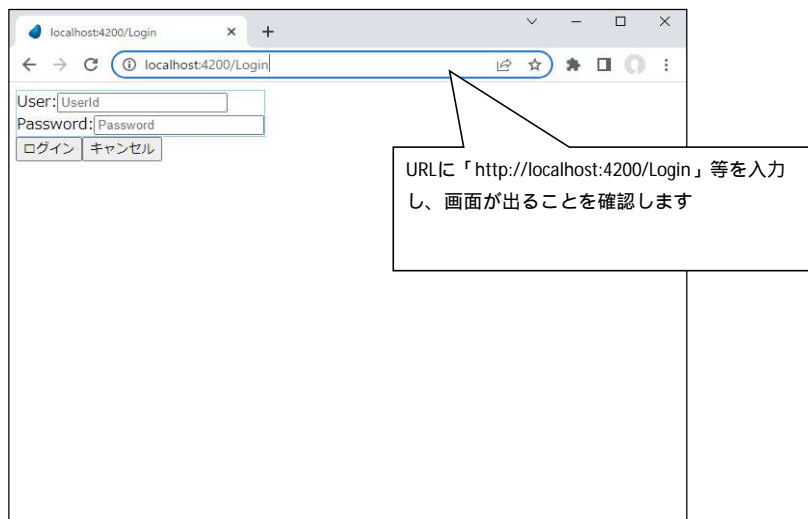


図2.2.4-2 Webブラウザでの動作確認

2.2.5 楽々ウェブクライアントの起動と初期設定

起動した実行エンジンは一旦停止しておきます。
登録したツールメニュー、もしくはツールバーのアイコンから「楽々ウェブクライアント」を起動します。
起動直後、下記の画面になります。

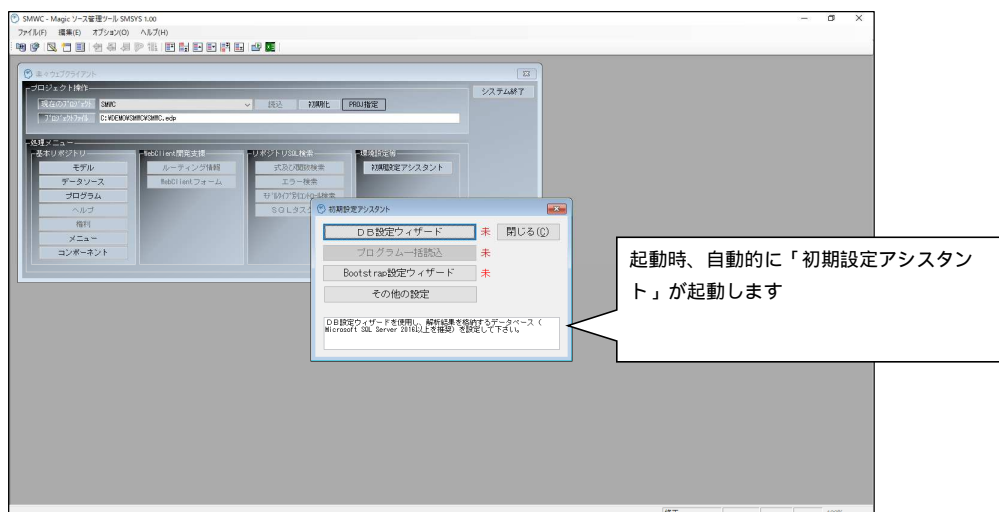


図2.2.5-1 楽々ウェブクライアント起動時に表示された初期設定アシスタント

イ) 初期設定アシスタント

楽々ウェブクライアントの実行環境セットアップのためのメニューを表示します。

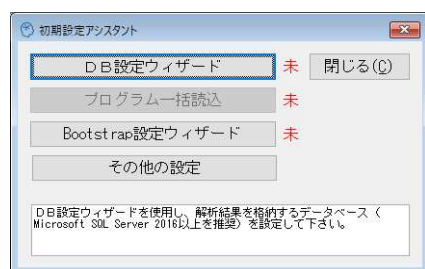


図2.2.5-2 初期設定アシスタント

次の場合、起動時に初期設定アシスタントをポップアップしません

- Webアプリケーションの設定でスタートアッププログラムが指定されていない (MAGIC xpa 2.x、uniPaaS等の場合を含む)
- DB設定でSQLite等の他のSQLゲートウェイが設定されている
- DB設定、プログラム解析、Bootstrapの初期設定が全て済んでいる

DB設定ウィザード

ソース解析結果とデザイン情報を格納するデータベースを設定します。
必ずプロジェクト単位に1データベースを設定するようにしてください。他のプロジェクトと一緒に共有することはできません。
サポートしているRDBMSは Microsoft SQL Server (2016以上のバージョン) です。

プログラム一括読み込み

WebClientタスクを持つプログラムソースを解析し、RDBSMに格納します。

Bootstrap設定ウィザード

BootstrapをAngular環境にインストールします。
予め用意してあるテンプレートを利用してスタートアッププログラムで表示するナビゲーションバーを作成することも可能です。

その他の設定

その他、動作環境に関連するオプション条件を設定します。

ロ) 「DB設定ウィザード」の起動

a) データベース・ゲートウェイの選択

DB設定ウィザード

データベース・ゲートウェイの選択

機能を有効にするためにはSQL Serverを選択します。

選択するデータベース

ゲートウェイ: Memory
SQLServer

現在の設定値は、「Memory」です。

WebClient開発支援機能を利用するには「SQL Server」を選択してください。

次へ(N) <戻る(B) 戻る(B) 次へ(N) >

「SQL Server」を選択し「次へ(N)」進みます

図2.2.5-3 DB設定ウィザード / データベース・ゲートウェイの選択画面

b) サーバーの選択

DB設定ウィザード

サーバーの選択

利用する Microsoft SQL Server と認証情報を設定します。

接続情報

サーバー名: ...

サーバーを入力して下さい。

ユーザー名:

パスワード:

WebClient開発支援機能を利用するには「Microsoft SQL Server 2016」以降のバージョンを選択してください。

次へ(N) <戻る(B) 戻る(B) 次へ(N) >

接続するSQL Serverのサーバー名とユーザ情報を選択し「次へ(N)」進みます
(Windows認証の場合は「ユーザー名」「パスワード」は省略可能)

Microsoft SQL Serverは2016以降のバージョンを選択します。

図2.2.5-4 DB設定ウィザード / サーバーの選択画面

c) データベースの選択

DB設定ウィザード

データベースの指定

解析データを格納するデータベースを指定します。

接続情報

サーバー: SURFACE-LT\SQLSERVER19

DB名: ...

DB名を入力して下さい。

新規のデータベース名を入力した場合は自動的に作成します。
既存のデータベースを選択するには「...」ボタンを押してください。
サーバーを変更するには「Back」ボタンで前の画面に戻ります。

【重要】
他のプロジェクトと同じデータベースを設定しないでください。
誤動作の原因となります。

次へ(N) <戻る(B) 戻る(B) 次へ(N) >

データベースの名称を指定します
新しい名称を入力した場合は新規に作成します
既存のデータベースを選択する場合は「...」ボタンで表示する一覧から選択します

図2.2.5-5 DB設定ウィザード / データベースの指定画面

d) 最終処理確認

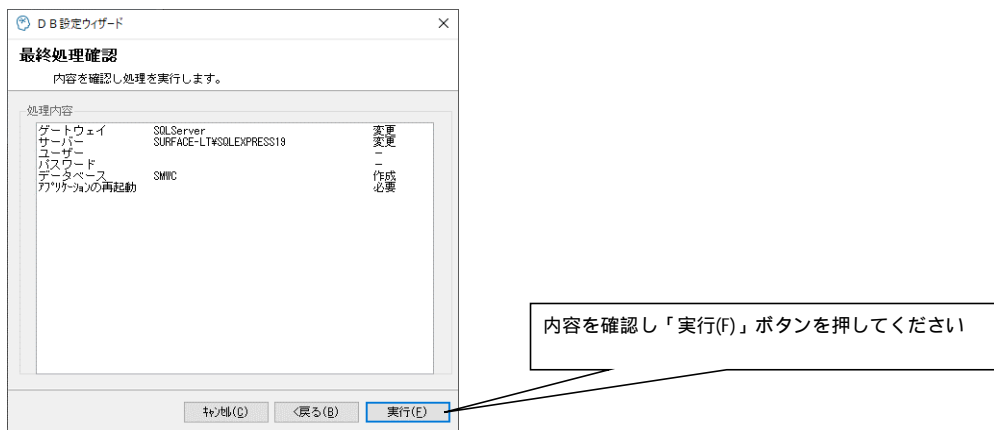


図2.2.5-6 DB設定ウィザード / 最終処理確認画面

データベースの設定処理が実行され、終了すると下記のダイアログを表示します。
「OK」ボタンを押すと再起動します。

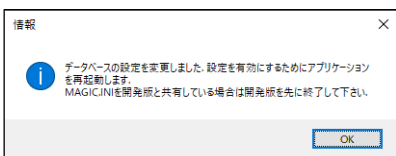


図2.2.5-7 DB設定ウィザード / 処理実行後のダイアログ画面

ハ) プログラム一括読み込み

再起動後、再び初期設定アシスタントが起動します。

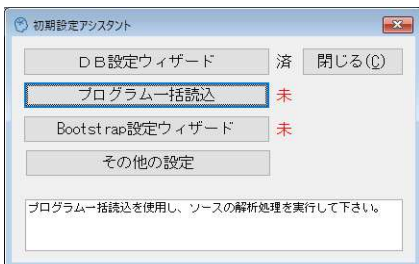


図2.2.5-8 再起動後の初期設定アシスタント

「プログラム一括読み込み」ボタンを押してプログラムを起動します。

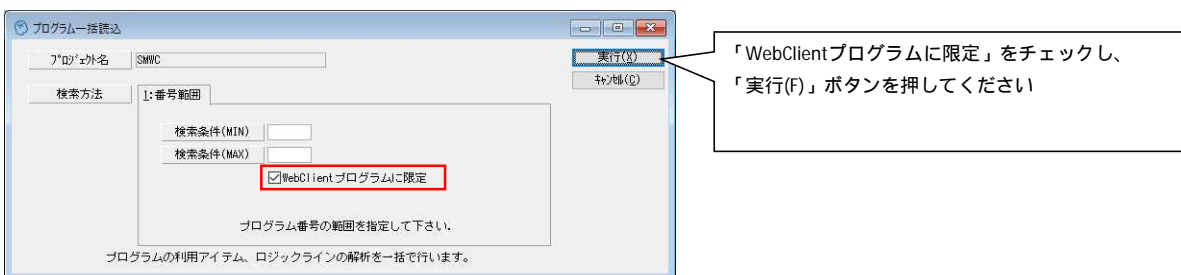


図2.2.5-9 プログラム一括読み込みによるプログラムの解析

ニ) Bootstrap設定ウィザード

処理が終了すると再び初期設定アシスタントに戻ります。
「Bootstrap設定ウィザード」を起動します。

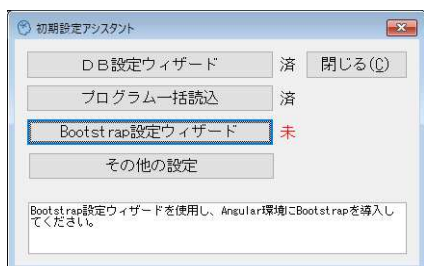
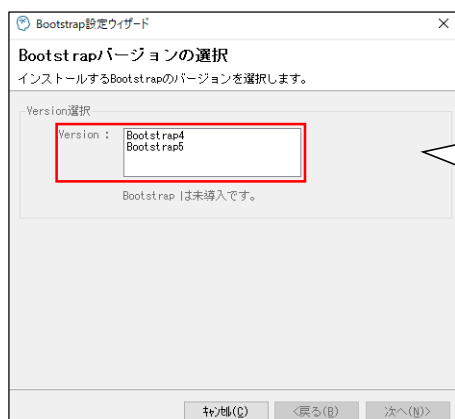


図2.2.5-10 プログラム一括読み込み後の初期設定アシスタント

a) Bootstrapバージョンの選択

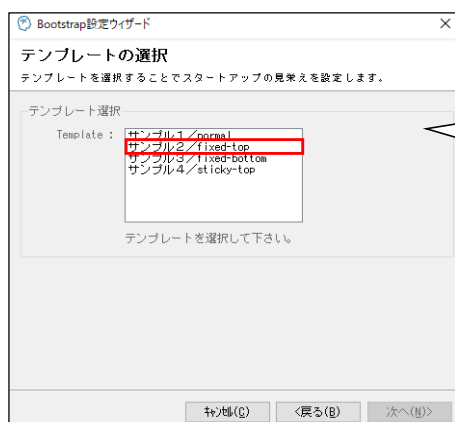


インストールするBootstrapのバージョンを選択します
差し当たっては「Bootstrap5」を選択してください
なお、インストール後も一旦除外することにより入れ替えが可能です
(図2.2.5-12)



図2.2.5-11 Bootstrap設定ウィザード / Bootstrapバージョンの選択

b) テンプレートの選択

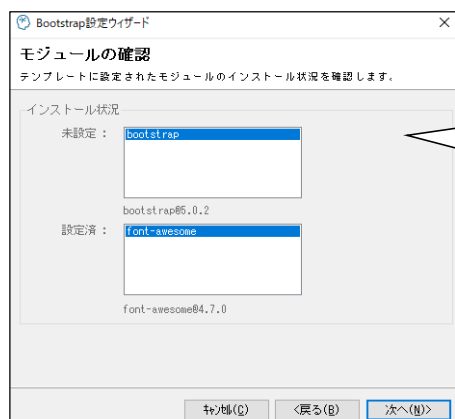


ナビゲーションバー用に準備されたテンプレートを選択します
ここでは「サンプル2 / fixed-top」を選択して下さい。

サンプル1 / normal	ナビゲーションバーを固定しません
サンプル2 / fixed-top	ナビゲーションバーを画面上部に固定します
サンプル3 / fixed-bottom	ナビゲーションバーを画面下部に固定します
サンプル4 / sticky-top	スクロールしたときナビゲーションバーを画面上部に固定します

図2.2.5-13 Bootstrap設定ウィザード / テンプレートの選択

c) モジュールの確認



インストール状況を確認します

未設定：今回、インストールされるモジュール
設定済み：既存のモジュール（インストールされません）

図2.2.5-14 Bootstrap設定ウィザード / モジュールの確認

d) ナビゲーションバーの設定

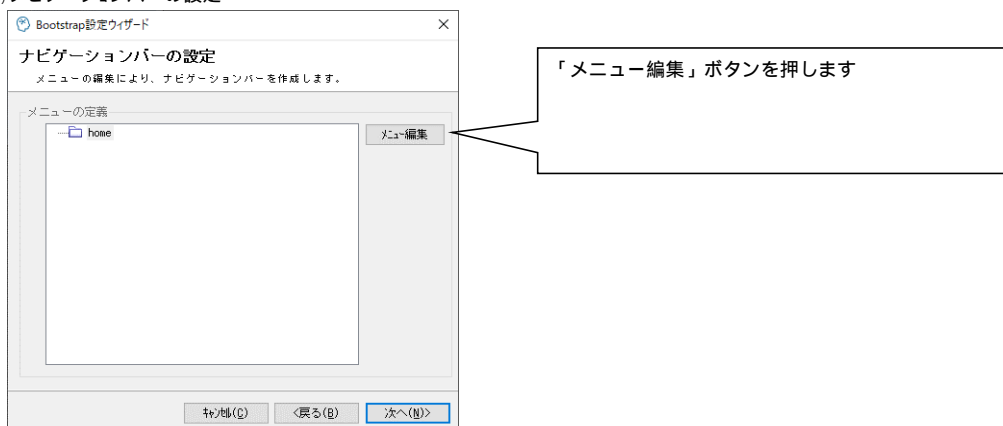


図2.2.5-15 Bootstrap設定ウィザード / ナビゲーションバーの設定

d-1) メニュー定義画面について

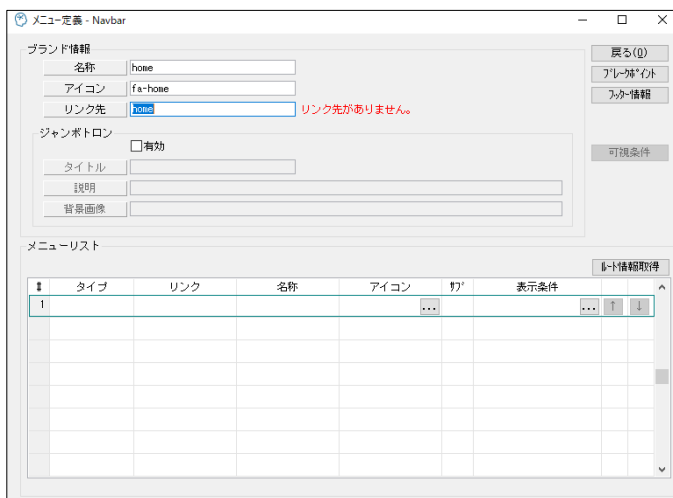


図2.2.5-16 Bootstrap設定ウィザード / ナビゲーションバーの設定 / メニュー定義プログラム (初期状態)

メニュー定義プログラムの設定項目

項目	説明
ブランド情報	ナビゲーションバーの情報を設定します。
名称	ナビゲーションバーの左端に表示するテキストを入力します。
アイコン	fontawesomeのアイコン名を選択します。 「F5:ズーム」キーで一覧表示から選択することができます。
リンク先	プログラムのルート名を入力します。 「リンク先がありません。」と表示されている場合は、正しいルート名ではありません。 「F5:ズーム」キーで一覧から選択します。
ジャンボトロン	ページの上部に配置する画像ファイルに関する情報を設定します。
有効	チェックすると有効に、チェックを外した状態では非表示にします。
タイトル	サイトのタイトルを入力します。
説明	説明文を入力します。
背景画像	表示する画像ファイルを入力します。 予め、表示する画像ファイルを「src%assets」配下にコピーしておいて下さい。 「F5:ズーム」キーでフォルダのファイル一覧から選択します。
ブレイクポイント	ブラウザの表示幅によって見栄えを変更させるための閾値(しきいち)を指定します。
フッタ情報	フッタに出力する会社名や、表示させるリンク(メニューリストで設定済みから選択)を設定
可視条件	ジャンボトロンの表示条件を設定します。 カスタムプロパティが設定されていない場合は無効状態になります。
メニューリスト	ナビゲーションバーに設定するメニューを定義します。 「ルート情報取得」ボタンで選択することも可能です。
#	表示順を表します。 行追加した直後は挿入位置に表示されます。 (必要に応じてカラムをクリックし再整列させて下さい)
タイプ	コンボボックスから下記の選択を行います。 S=サブメニュー サブメニューを設定する場合に選択します。 (設定できるのは最上位の階層のみです) L=リンク ルートで設定したプログラムをメニューにします。 D=区切り 区切りを挿入。最上位の階層の場合は挿入位置で左右に分離します。
リンク	プログラムのルート名を入力します。 「F5:ズーム」キーで一覧から選択します。
名称	メニューに表示させるテキストを設定します。
アイコン	アイコン名を入力します。 「F5:ズーム」キーで一覧表示から選択することができます。
サブ	タイプで「S=サブメニュー」を選択したときは「編集」ボタンを表示します。
表示条件	メニューの表示条件を指定することが可能です。 「F5:ズーム」キーで一覧から選択します。
	ボタンを押すことにより表示順を変更します。

以下、基本的なナビゲーションバーを作成する手順を示します。

d-2) ブランド情報

ブランド情報

名称:

アイコン: fa-home

リンク先: home

ジャンボトロン

有効 ☐

タイトル:

説明:

背景画像:

メニューリスト

#	タイプ	リンク
1		

ルート選択

ルート	PR	プログラム名
DataDetail	15	DataDetail
DataList	14	DataList
GridTest	20	GridTest
Home	6	Home
Login	7	Login
LoginOverlay	10	LoginOverlay
Logout	8	Logout
ModelList	16	ModelList
Program	13	Program
ProgramList	12	ProgramList
Repository	9	Repository
ZoomButtonTest	19	ZoomButtonTest

- 「名称」を省略するために空白にします
- アイコンが予め設定されている「fa-home」をそのまま使用します
- リンク先は誤ったルート名になっているため、項目でズームし、「Home」を選択します
- ジャンボトロンは今回は使用しないので、そのままにします

図2.2.5-17 Bootstrap設定ウィザード / ナビゲーションバーの設定 / メニュー定義プログラム / ブランド情報

d-3) メニューリスト

ブランド情報

名称:

アイコン: fa-home

リンク先: Home

ジャンボトロン

有効 ☐

タイトル:

説明:

背景画像:

メニューリスト

#	タイプ	リンク	名称	アイコン	表示条件
1	L-Link

ルート情報取得

編集

- 画面下部のメニューリストに移動します
- タイプに「L-Link」が設定されますが、これを「S-サブメニュー」に変更します
- 名称に「Repository」を入力します
- 表示されたサブメニュー「編集」ボタンを押します

図2.2.5-18 Bootstrap設定ウィザード / ナビゲーションバーの設定 / メニュー定義プログラム / メニューリスト(1)

d-4) サブメニューの編集

メニュー定義 - Repository

タイプ: L-Link

リンク: ...

名称: ...

アイコン: ...

表示条件: ...

ルート情報取得

編集

ルート選択

選択	行#	コール名	番号	コール先名	ルート名	出力先	済
<input type="checkbox"/>	4	P=7"0"0"0"0"	6	Home	Home		<input type="checkbox"/>
<input type="checkbox"/>	5	P=7"0"0"0"0"	7	Login	Login		<input type="checkbox"/>
<input type="checkbox"/>	6	P=7"0"0"0"0"	8	Logout	Logout		<input type="checkbox"/>
<input type="checkbox"/>	7	P=7"0"0"0"0"	8	Repository	Repository		<input type="checkbox"/>
<input checked="" type="checkbox"/>	8	P=7"0"0"0"0"	12	ProgramList	ProgramList		<input type="checkbox"/>
<input type="checkbox"/>	9	P=7"0"0"0"0"	13	Program	Program		<input type="checkbox"/>
<input checked="" type="checkbox"/>	10	P=7"0"0"0"0"	14	DataList	DataList		<input type="checkbox"/>
<input type="checkbox"/>	11	P=7"0"0"0"0"	15	DataDetail	DataDetail		<input type="checkbox"/>
<input checked="" type="checkbox"/>	12	P=7"0"0"0"0"	16	ModelList	ModelList		<input type="checkbox"/>
<input checked="" type="checkbox"/>	13	P=7"0"0"0"0"	19	ZoomButtonTest	ZoomButtonTest		<input type="checkbox"/>
<input checked="" type="checkbox"/>	14	P=7"0"0"0"0"	20	GridTest	GridTest		<input type="checkbox"/>

- 「ルート情報取得」ボタンを押します
- 表示されたルート選択画面で図のように追加するルート名にチェックを入れます
- 「OK」ボタンを押して閉じます

図2.2.5-19 Bootstrap設定ウィザード / ナビゲーションバーの設定 / メニュー定義プログラム / サブメニュー編集(1)

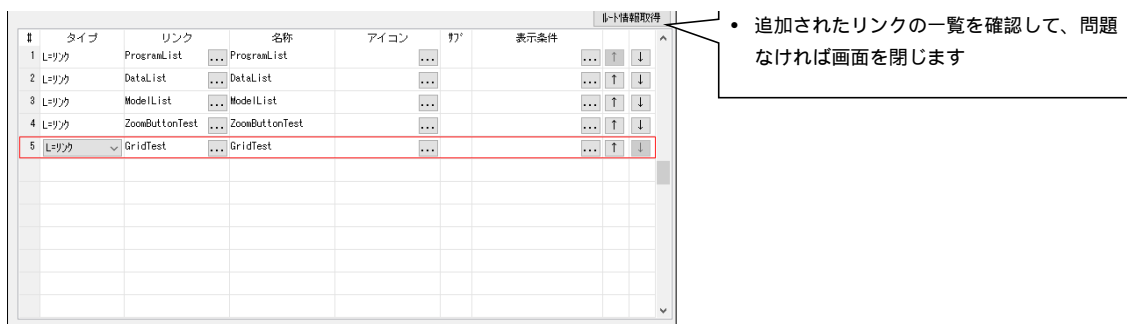


図2.2.5-20 Bootstrap設定ウィザード / ナビゲーションバーの設定 / メニュー定義プログラム / サブメニュー編集(2)

d-5) 区切り、ログインログオフ用コールの追加

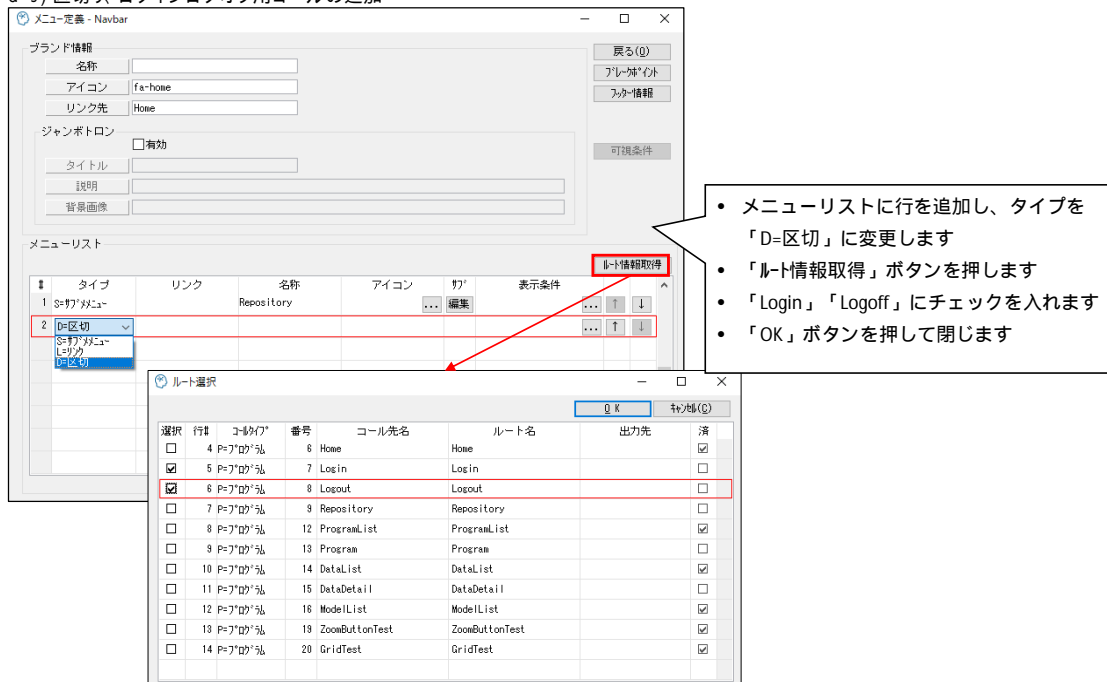


図2.2.5-21 Bootstrap設定ウィザード / ナビゲーションバーの設定 / メニュー定義プログラム / メニューリスト(2)

d-6) 表示条件の設定

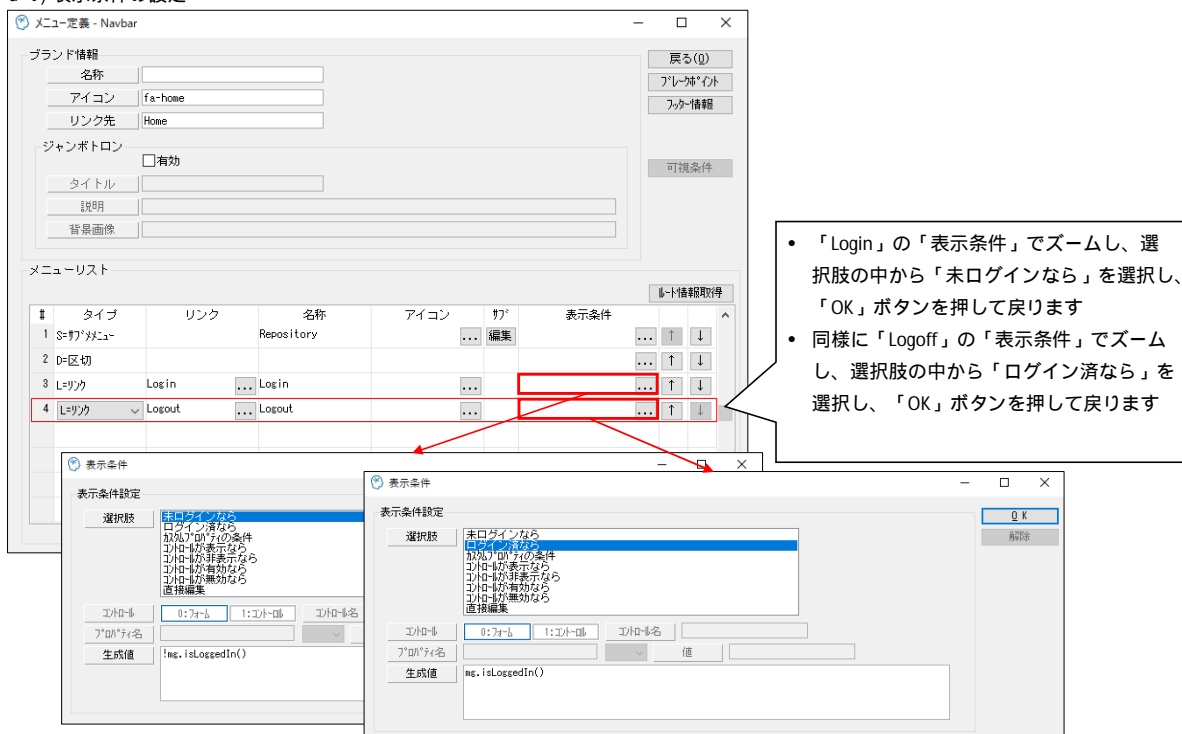


図2.2.5-22 Bootstrap設定ウィザード / ナビゲーションバーの設定 / メニュー定義プログラム / メニューリスト(3)

d-7) メニュー定義を完了して戻る

編集が終わったら「戻る」ボタンでウィザードに戻ります。

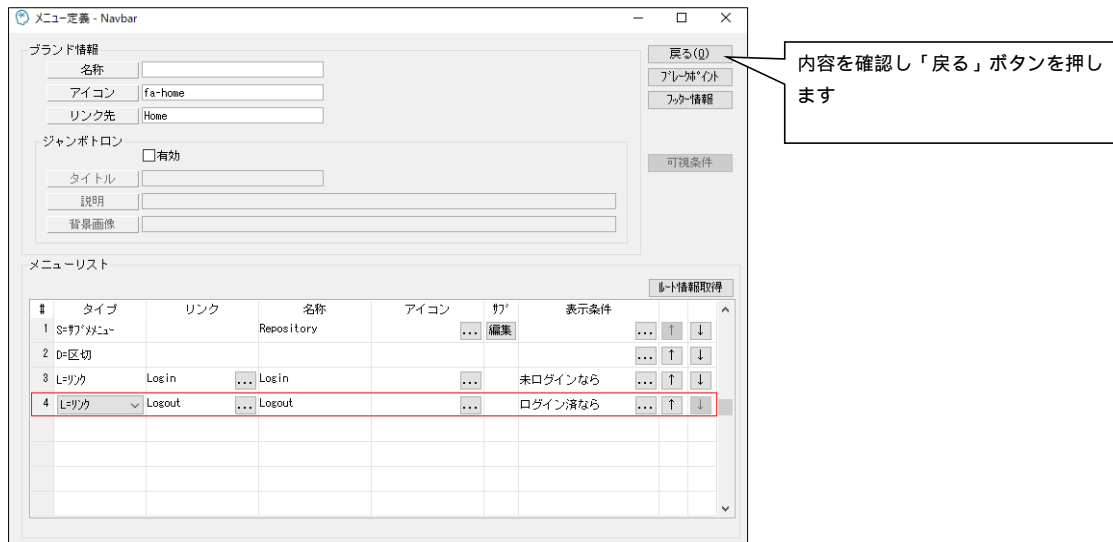


図2.2.5-23 Bootstrap設定ウィザード / ナビゲーションバーの設定 / メニュー定義プログラム (完了)

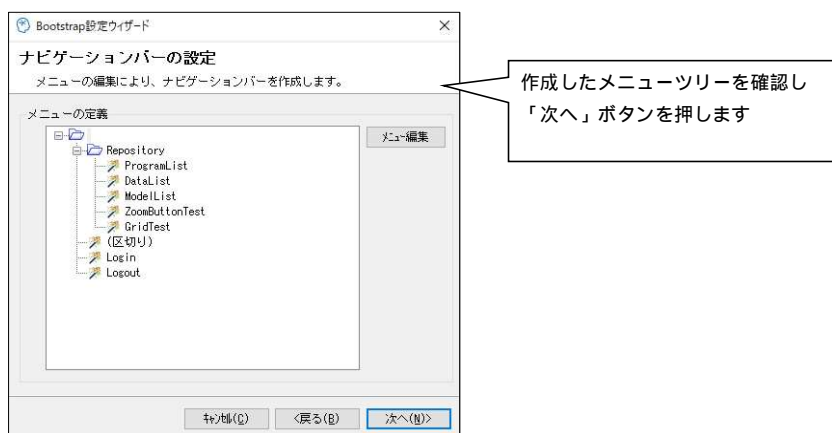


図2.2.5-24 Bootstrap設定ウィザード / ナビゲーションバーの設定 (完了)

e) 最終処理確認

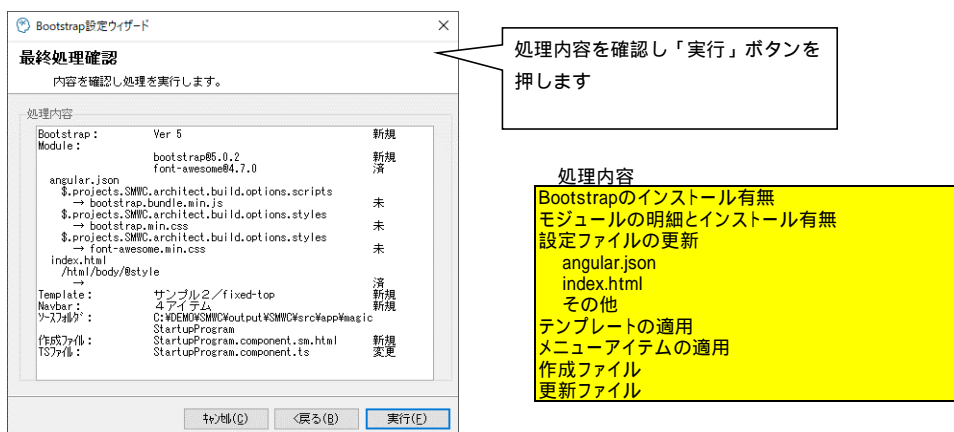


図2.2.5-25 Bootstrap設定ウィザード / 最終処理確認

実行後、下記のダイアログを表示します。

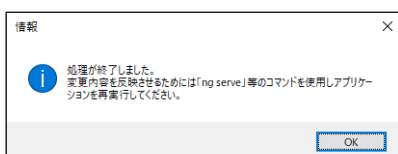


図2.2.5-26 処理正常終了時のメッセージ

Bootstrapを新規にインストールした場合は、ターミナル画面から「ng serve --open」等を入力し、サーバを起動しなさい

f) 動作確認

f - 1) ブラウザ表示

MAGICの実行エンジンを起動(「デバッグ」「プロジェクトの実行」)します。
Angularのサーバーが実行中であれば再起動します。
ブラウザを起動します。
正常に動作した場合は、下記のような画面を表示します。

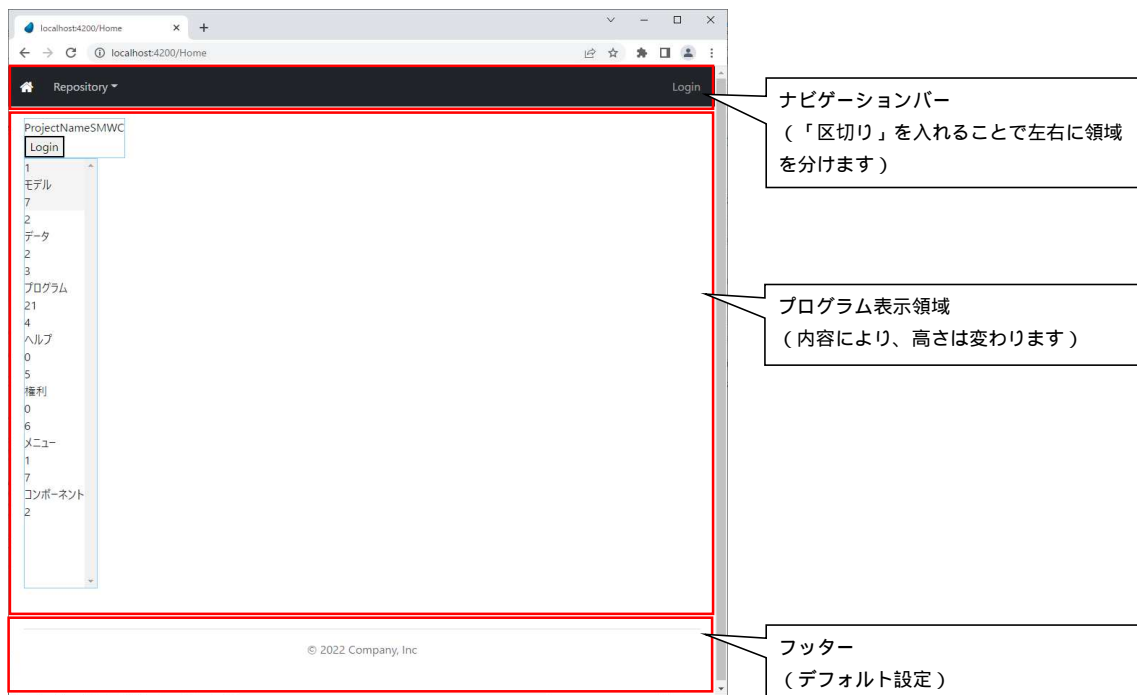


図2.2.5-27 作成されたスタートアッププログラム(ナビゲーションバー)

f - 2) Visual Source Codeでファイルを確認

Visual Source Codeで「SMWC*src*app*magic*StartupProgram*StartupProgram.component.sm.html」を開きます。

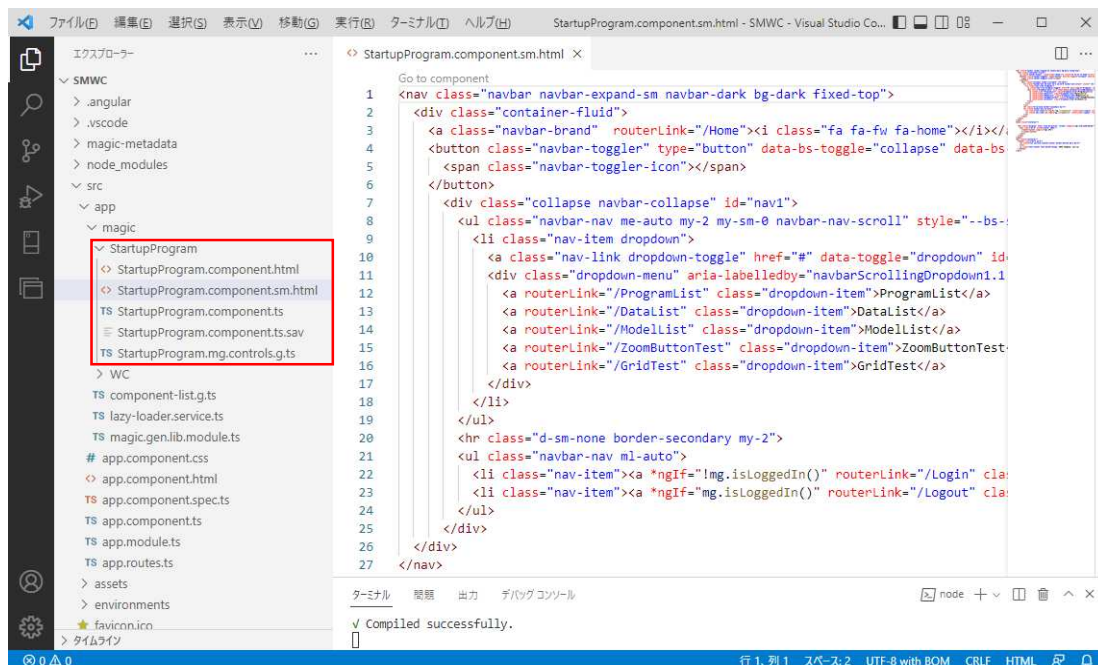


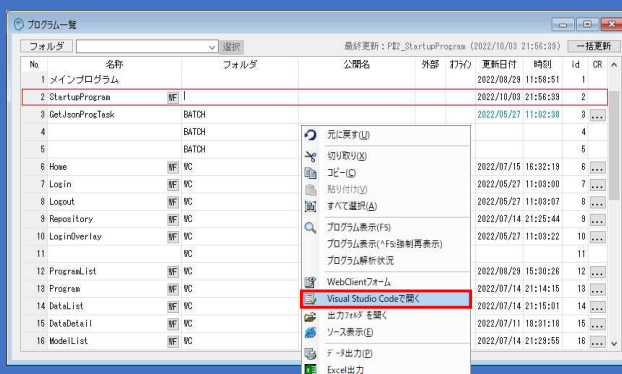
図2.2.5-28 Angularプロジェクトに作成されるファイルの確認

フォルダ内のファイル

ファイル名	説 明
*.component.html	MAGICが生成したTemplateファイル
*.component.sm.html	楽々ウェブクライアントが追加したTemplateファイル
*.component.ts	楽々ウェブクライアントが更新したTSファイル
*.component.ts.sav	楽々ウェブクライアントが更新したTSファイルのバックアップファイル
*.component.controls.g.ts	MAGICが生成したcontrols..tsファイル

< 補足 > Visual Source Codeで目的のソースを開く方法

楽々ウェブクライアントでは、いたるところから Visual Source Code を開けるようにしています。
例えば、プログラム一覧から起動するには、コンテキストメニューを開き、「Visual Studio Codeで開く」を選択して下さい。



他にも、WebClientフォーム画面の上部にある「Vs Codeで開く」ボタンや、「テンプレートの変換」プログラムのコントロールのコンテキストメニューから呼び出すことが可能です。

補足1) ナビゲーションバー / サンプルテンプレートについて

テンプレートを利用しナビゲーションバーのスタイルを変更することができます。

サンプル1 / normal	ナビゲーションバーを固定しません
サンプル2 / fixed-top	ナビゲーションバーを画面上部に固定します
サンプル3 / fixed-bottom	ナビゲーションバーを画面下部に固定します
サンプル4 / sticky-top	スクロールしたときナビゲーションバーを画面上部に固定します

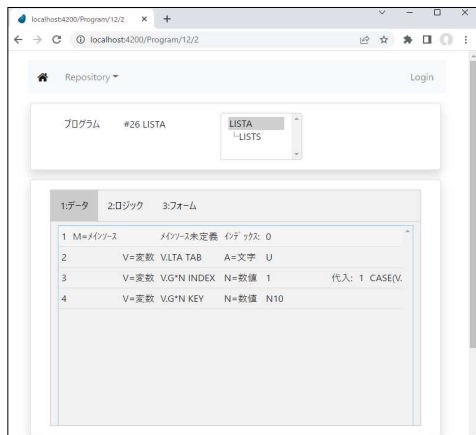


図2.2.5-29 「サンプル1 / normal」

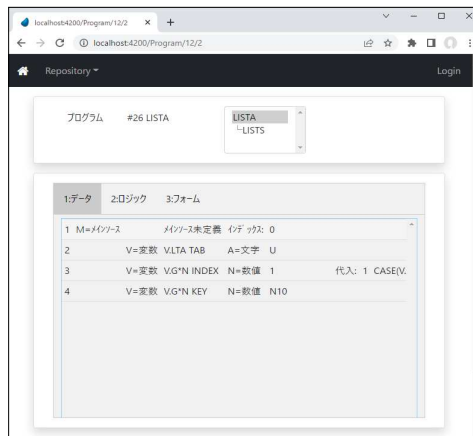


図2.2.5-30 「サンプル2 / fixed-top」

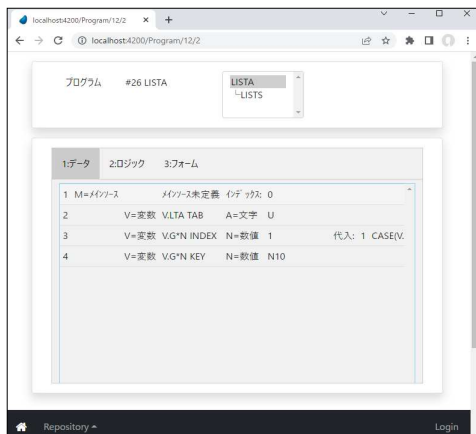


図2.2.5-31 「サンプル3 / fixed-bottom」

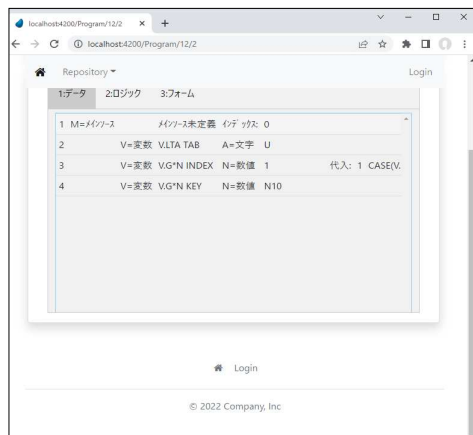


図2.2.5-32 「サンプル4 / sticky-top」

補足2) ナビゲーションバー / ブレークポイントの設定について

「ブレークポイント」ボタンでナビゲーションバーを簡易表示させる閾値(しきいち)を指定することが可能です。

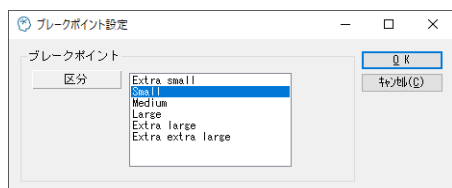


図2.2.5-32 ブレークポイント設定

ブレークポイント		
Extra small	< 576px	-
Small	576px	sm (初期値)
Medium	768px	md
Large	992px	lg
Extra large	1200px	xl

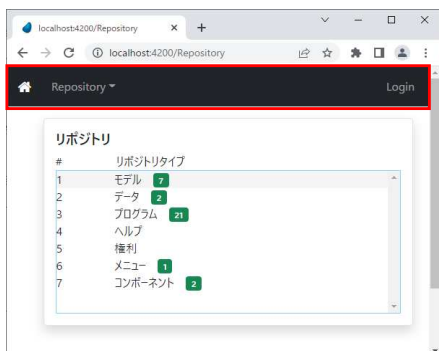


図2.2.5-33 ウィンドウの幅が閾値より大きい場合

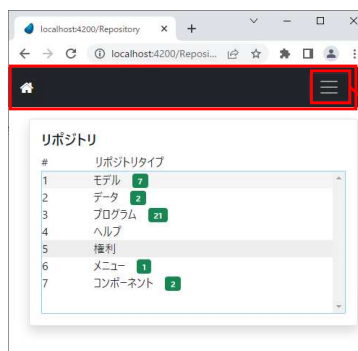
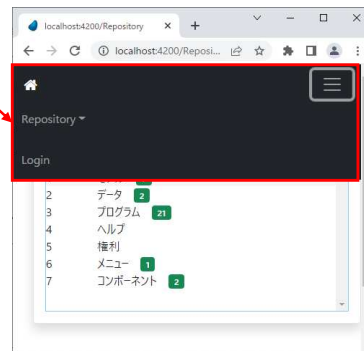


図2.2.5-34 ウィンドウの幅が閾値より小さい場合



補足3) ナビゲーションバー / フッタの設定について

メニュー定義画面の「フッター情報」ボタンからフッタの設定が可能です。

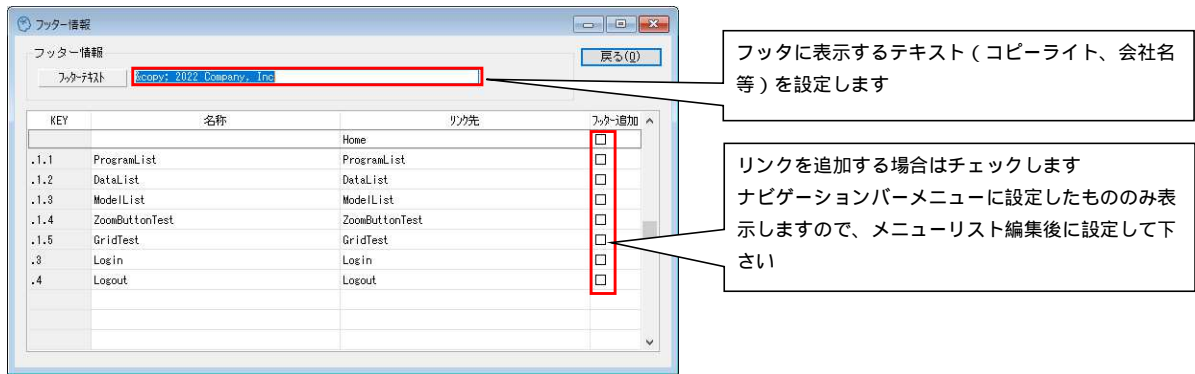


図2.2.5-35 フッター情報

設定例)

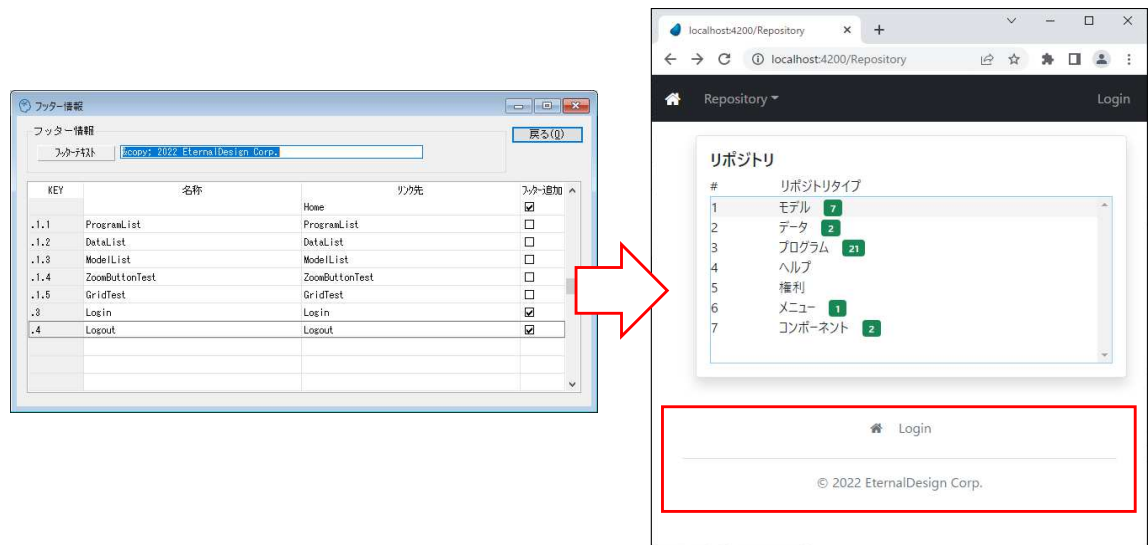


図2.2.5-36 フッター情報の設定例

補足4) ナビゲーションバー / ジャンボトロン

ジャンボトロンを設定すると、画面上部に画像を表示します。

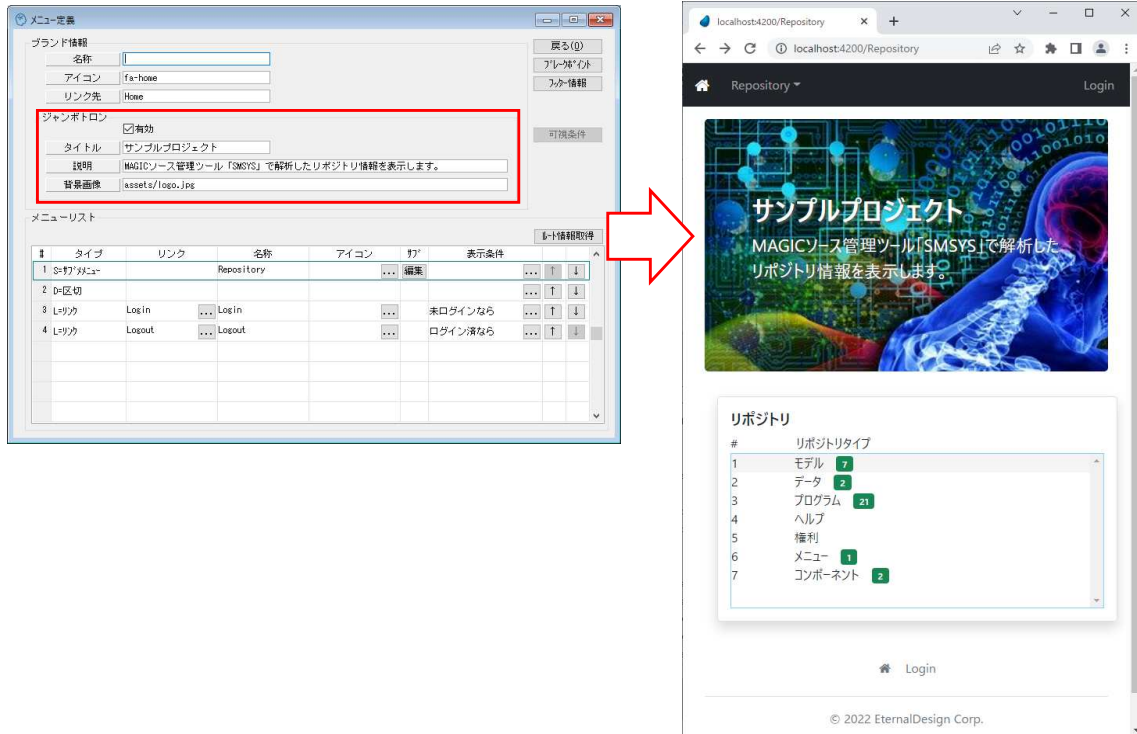


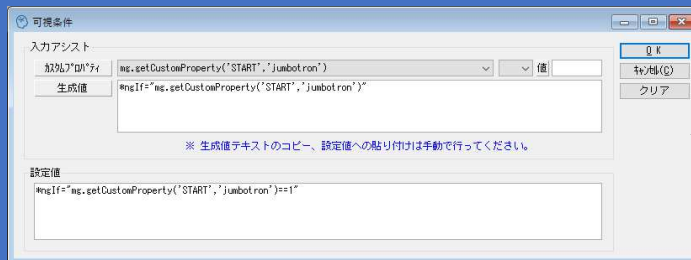
図2.2.5-37 ジャンボトロンの設定例

< 補足 >

背景画像は「assets」フォルダからの相対パスで指定して下さい。

「可視条件」ボタンは、ジャンボトロンの表示 / 非表示を制御します。

(カスタムプロパティが設定されているときに有効になります)



ジャンボトロンの可視条件を設定します
カスタムプロパティに対する式を設定して下さい

ホ) その他の設定



図2.2.5-32 Bootstrap設定ウィザード終了後の初期設定アシスタント

「その他の設定」を起動します。
 楽々ウェブクライアントに関連する設定箇所を下図に赤枠で囲って表示します。
 特に付加識別文字と拡張子の設定は、全プロジェクト共通なので、注意して設定して下さい。(途中で変更することはあまり推奨されません)

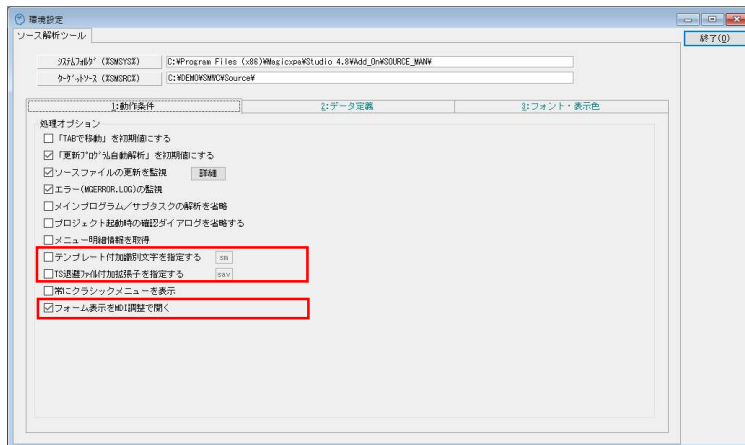


図2.2.5-33 環境設定画面

動作条件 / 設定項目	
項目	説明
テンプレート付加識別文字を指定する	テンプレートの変換プログラムで作成するファイルを「(フォーム名).component.sm.html」の様に「.sm」を挿入して作成します。その識別文字「sm」を変更することが可能です。
TS退避ファイル付加拡張子を指定する	テンプレートの変換プログラムで書き変えたTSファイルの退避ファイルを「(フォーム名).component.ts.sav」の様に「.sav」を拡張子として作成します。その拡張子「sav」を変更することが可能です。
フォーム表示をMDI調整で開く	テンプレートの変換プログラムから「フォーム確認」ボタンでコントロールの一覧を表示したとき、画面をMDI全体に表示するかどうかを指定します。

「ソースファイルの更新を監視」について

楽々ウェブクライアントはMAGIC xpa Ver4.xの開発環境と並行しての運用を前提としています。
 プログラム単位での更新時刻の変更を検知した場合は、それをもってアラートする仕組みを持っています (SMSYSの基本機能)。

No	名称	フォルダ	公開名	外部	タイプ	更新日付	時刻	id	DR
1	メインプログラム					2022/04/29	10:21:24	1	
2	StartupProgram	WFE				2022/05/29	10:05:04	2	
3	GetJsonProgTask	BATCH				2022/05/27	11:02:38	3	...
4		BATCH						4	
5		BATCH						5	
6	Home	WFE	WC			2022/05/27	11:02:50	6	...
7	Login	WFE	WC			2022/05/27	11:03:00	7	...
8	Logout	WFE	WC			2022/05/27	11:03:07	8	...
9	Repository	WFE	WC			2022/05/27	11:03:15	9	...
10	LoginOverlay	WFE	WC			2022/05/27	11:03:22	10	...
11	ProgramList	WFE	WC			2022/05/27	11:03:29	11	...
12	Program	WFE	WC			2022/05/27	11:04:05	12	...
13	DataList	WFE	WC			2022/06/06	16:28:28	13	
14			WC					14	
15			WC					15	
16			WC					16	

図2.2.5-34 プログラム一覧画面による更新監視結果の表示

楽々ウェブクライアントではAngularソースの更新にあたってはMAGICのソース変更を都度チェックしていますので、必ずしも上記の設定が必須というわけではありません。
 MAGICのソース変更を検知した場合は「テンプレートの変換」プログラム起動時に自動的に再取得します。

2.3 楽々ウェブクライアントの基本機能

2.3.1 処理メニュー

処理メニューは用意された様々なプログラムを起動するためのメニューです。その基本操作を説明します。



図2.3.1-1 楽々ウェブクライアントの初期メニュー (処理メニュー)

処理メニュー	
項目	説明
プロジェクト操作	プロジェクトに関する表示、変更操作等を行います。
現在のプロジェクト	実行中のプロジェクト名を表示します。 複数のプロジェクトが登録済みの場合は、コンボボックスを開くことにより、他のプロジェクトに変更できます。
「読み込」ボタン	選択したプロジェクトを読み込むためのボタン。 実際はプロジェクト単位に解析データ用のDBを設定して運用するため、コンボボックスを変更しようとした時点で確認用のメッセージボックスが開きます。(図2.3.1-2) DBを設定していないプロジェクト同士を切り替えるときはこのボタンを使用して下さい。
「初期化」ボタン	解析データを初期化します。 但し、WebClientフォームの編集情報は初期化の対象外です。必要に応じてフォーム単位に個別に指定します。(2.4.6「その他の機能 / 変更を戻す」を参照)
プロジェクトファイル	現在読み込んでいるプロジェクトのEDPファイルをフルパスで表示します。
コンテキストメニュー	マウス右ボタンで表示するメニュー (図2.3.1-3)
プロジェクト特性	プロジェクトファイルの概要を表示します。(図2.3.1-4)
プロジェクト追加	任意のプロジェクトを追加することが可能です。 ファイルを開くダイアログからEDPファイルを選択します。
プロジェクトフォルダを開く	エクスプローラーでEDPファイルのあるフォルダを開きます。
MGERROR.LOGを開く	プロジェクトフォルダのエラーファイルを開きます。
初期設定アシスタント	初期設定アシスタントを起動します。
メニューセットを変更	ソース管理ツール「SMSYS」のデフォルトのメニューに変更します。(図2.3.1-5)
処理メニュー	各種プログラムを起動するためのメニューを表示します。
基本リポジトリ	MAGICの基本リポジトリを確認するためのメニューを表示します。
モデル	モデルの一覧を表示します。
データ	データの一覧を表示します。
プログラム	プログラムの一覧を表示します。(詳細は、2.3.4を参照)
ヘルプ	ヘルプの一覧を表示します。
権利	権利の一覧を表示します。
メニュー	メニューの一覧を表示します。
コンポーネント	コンポーネントの一覧を表示します。またコンポーネント毎のアイテムの一覧を表示します。
WebClient開発支援	WebClient関連のメニューを表示します。
ルーティング情報	スタートアッププログラムに設定したプログラムのルート情報を表示します。 またルートで呼び出している箇所の確認が可能です。(詳細は、2.3.2を参照)
WebClientフォーム	WebClientフォームの一覧を表示します。(詳細は、2.3.3を参照)
リポジトリSQL検索	SQLのクエリを使用した検索プログラムのメニューを表示します。
式及び関数検索	式で使用している文字列を指定して該当箇所の検索を行います。
エラー検索	エラーコマンド設定箇所の検索を行います。
SQLタスク検索	SQLのクエリを使用したプログラムを検索します。
環境設定等	環境設定に関連するメニューを表示します。
初期設定アシスタント	初期設定アシスタントを起動します。
「システム終了」ボタン	アプリケーションを終了します。

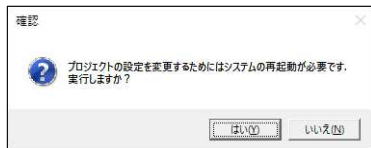


図2.3.1-2 プロジェクト変更時のメッセージ

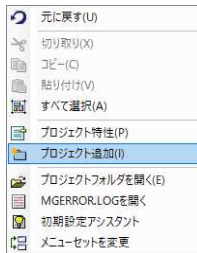


図2.3.1-3 コンテキストメニュー

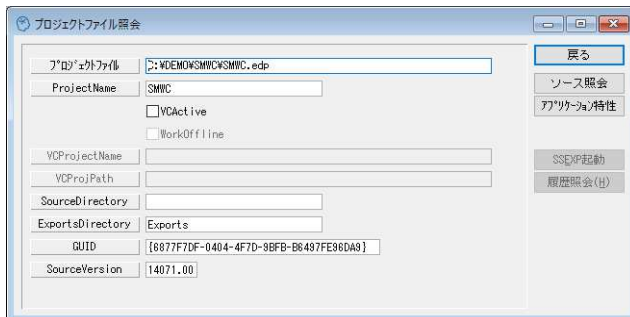


図2.3.1-4 プロジェクトファイル照会



図2.3.1-5 メニューセット切り替えにより表示されるメニュー(クラシックメニュー)

次の場合、クラシックメニューで表示します。

Webアプリケーションの設定でスタートアッププログラムが指定されていない (MAGIC xpa 3.x、uniPaaS等の場合を含む)

DB設定でSQLite等の他のSQLゲートウェイが設定されている

環境設定で「常にクラシックメニューを表示」がチェックされている

WebClientプロジェクトの場合のみ、メニューから選択して切り替えが可能

2.3.2 ルーティング情報

スタートアッププログラムに設定されたルートイベントとそのプログラムコールによってAngularのルーティング情報が生成されます。また、個別のプログラムからルート呼出しが可能です。
「ルーティング情報」は、解析済みのWebClientプログラムからルーティングに関する情報を集約的に表示します。

1) 「ルーティング情報」の画面と項目

ルーティング情報

スタートアッププログラム/ルート定義

プログラム名: StartupProgram P#: 2 フォーム名: SF1 サブフォーム名: SF1 デフォルトアウトレット: ☒

ロジック: イベント/ルート 行#: 1 更新時刻: 2022/05/27 11:02

ルート定義明細

行#	コールタイプ	番号	コール先名	ルート名	出力先	式#	条件	フォルダ	最終更新時刻
4	P=プログラム	8	Home	Home				WC	2022/05/27 11:02
5	P=プログラム	7	Login	Login				WC	2022/05/27 11:03
6	P=プログラム	8	Logout	Logout		1	UserName<>' '	WC	2022/05/27 11:03
7	P=プログラム	8	Repository	Repository				WC	2022/05/27 11:03
8	P=プログラム	11	ProgramList	ProgramList				WC	2022/05/27 11:03
9	P=プログラム	12	Program [2行分]	Program				WC	2022/05/27 11:04

全フォーム(プログラム)

T#	タスク名	最終更新時刻	F#	フォーム名	CP	CV	ウインドウタイプ	テーブル/Webスタイル	処理
1		2022/05/27 11:03	2	ProgramList			D=デフォルト		<input type="checkbox"/> 変換
2	ProgramTbl	2022/05/11 16:20	3	ProgramTbl			D=デフォルト	P=繰り返し可能な項目	<input type="checkbox"/> 変換

Navbar作成
VS Codeで開く
ルート呼出箇所
カスタムプロパティ
カスタムバリデータ

図2.3.2-1 ルーティング情報

項目	説明
スタートアッププログラム	スタートアッププログラムに関する情報を表示します。
プログラム名	スタートアッププログラムの名称を表示します。
P#	スタートアッププログラムの番号を表示します。
ロジック	ロジックのイベントを表示します。
行#	ロジックのイベントの行番号を表示します。
フォーム名	フォーム名を表示します。
サブフォーム名	配置されたサブフォームのコントロール名を表示します。
デフォルトアウトレット	デフォルトアウトレットに設定されている場合はチェックされます。
更新時刻	スタートアッププログラムの最終更新時刻を表示します。
ルート定義明細	ルートイベントに設定されているプログラムコールの一覧を表示します。
行#	スタートアッププログラム内のロジック行番号を表示します。
コールタイプ	コールタイプ(P=プログラム / T=タスク)を表示します。
番号	コールタイプ別に番号(プログラム番号 / タスク番号)を表示します。
コール先名	コールタイプ別に名称(プログラム名 / タスク名)を表示します。
ルート名	ルート名を表示します。
出力先	出力先のサブフォームが指定されている場合はそのコントロール名を表示します。
式#	条件式が設定されている場合の式番号を表示します。
条件	条件式が設定されている場合の式内容を表示します。
フォルダ	プログラムのフォルダ名を表示します。
最終更新時刻	プログラムの最終更新時刻を表示します。
全フォーム(プログラム)	プログラムの全タスクの全フォームを表示します。
T#	タスク番号を表示します。
タスク#	タスクの階層を表す番号を表示します。
タスク名	タスク名を表示します。
F#	フォー番号を表示します。
フォーム名	フォーム名を表示します。
CP	カスタムプロパティの有無を表示します。表示される「…」ボタンで明細を確認できます。
CV	カスタムバリデータの有無を表示します。表示される「…」ボタンで明細を確認できます。
ウインドウタイプ	ウインドウタイプ(D=デフォルト / O=オーバーレイ)を表示します。
テーブル / Webスタイル	テーブルが存在するとき、Webスタイル(R=繰り返し可能な項目 / M=マテリアルデザイングリッド / H=HTMLテーブル)を表示します。
処理	Angularソースの編集(テンプレートの変換)を行う場合は「変換」ボタンを押します。編集したデータがある場合はチェックボックスにチェックが入ります。変換処理の詳細については「2.4 テンプレートの変換」を参照してください。
「Navbar作成」ボタン	Navbar作成プログラムをコールします。(2.3.2(2)「Navbar作成」の画面と項目を参照)
「VS Codeで開く」ボタン	テンプレートの選択、Navメニューの編集等、Bootstrap設定ウィザードと同様の更新が可能
「ルート呼出箇所」ボタン	Visual Source CodeでAngularソース(現在のTSファイルで指定されたtemplateUrl)を開きます。
「カスタムプロパティ」ボタン	全プロジェクト内のイベント実行「ルート」の箇所一覧を表示します。(図2.3.2-2)
「カスタムバリデータ」ボタン	全プロジェクト内のカスタムプロパティ設定箇所の一覧を表示します。(図2.3.2-3)
「カスタムバリデータ」ボタン	全プロジェクト内のカスタムバリデータ設定箇所の一覧を表示します。

P#	プログラム名	T#	タスク名	行#	処理名	F#	フォーム名	C#	コントロール名	項目	式	説明(ルット名)	スタック	項目	式	説明(アット名)	スタック
7	Login			4	E=イベント/ Login					??	4	'Home'	<input type="checkbox"/>	??			<input checked="" type="checkbox"/>
7	Login			8	E=イベント/ Login					??	5	'Login'	<input type="checkbox"/>	??			<input checked="" type="checkbox"/>
7	Login			12	E=イベント/ Dancel					??	4	'Home'	<input type="checkbox"/>	??			<input checked="" type="checkbox"/>
8	Logout			4	T=タスク/ P=前					??	3	'Login'	<input type="checkbox"/>	??			<input checked="" type="checkbox"/>
10	LoginOverlay			4	E=イベント/ Login					??	4	'Home'	<input type="checkbox"/>	??			<input checked="" type="checkbox"/>
10	LoginOverlay			8	E=イベント/ Login					??	5	'Login'	<input type="checkbox"/>	??			<input checked="" type="checkbox"/>
10	LoginOverlay			12	E=イベント/ Dancel					??	4	'Home'	<input type="checkbox"/>	??			<input checked="" type="checkbox"/>
11	ProgramList	.1	ProgramTbl	7	E=イベント/ タブ切					??	5	'Program'	<input type="checkbox"/>	??			<input checked="" type="checkbox"/>
13	DataList	.1	DataTbl	6	E=イベント/ タブ切					??	5	'Data'	<input type="checkbox"/>	??			<input checked="" type="checkbox"/>

図2.3.2-2 ルート呼出し箇所

名前	P#	プログラム名	タスク番号	タスク名	F#	フォーム名	C#	コントロール名	式#	式
count	9	Repository			2	Repository	6	name	2	Trim(Str(C,'52'))

図2.3.2-3 カスタムプロパティ

2) 「Navbar作成」の画面と項目

「Navbar作成」ボタンを押すと下記の画面を開きます。

図2.3.2-4 Navbar作成

Navbar作成の主要項目		
項目	説明	
利用するテンプレート	選択するテンプレートやその状態を表示します。	
テンプレートNo.	テンプレートを変更する場合はここでその番号を入力します。(「F5」キーでズーム)	
ファイル(*.html)の作成	テンプレートに関する状態を表示します。	
作成ファイル	オリジナルのファイルではなく、別なファイルに書き変えた結果を出力します。	
ファイル作成	「作成済 / 新規」で状態を表示します。	
TSファイルの定義	TSファイルに関する状態を表示します。	
現在のTemplate定義	TSファイル中、「templateUrl」で指定されているファイル名を表示します。	
変更するTemplate定義	当プログラムが変更しようとしているファイル名を表示 (現在値と同一の場合は無効表示)	
TSファイル更新	「変更済 / 相違」で状態を表示します。	
「Navbarメニュー編集」ボタン	Navbarメニューを編集します。詳細は「2.2.5 目」を参照	
「ファイルをバックアップ」ボタン	TSファイルをエディタ等で直接編集したときはボタンが有効になります。バックアップすることで、「*.sav.ts」に保存します。	

3) スタートアッププログラム編集時の操作

スタートアッププログラムにルーティングプログラムを追加したときのナビゲーションバー編集手順について説明します。

例として、開発版で、プログラム番号13の「DataList」をスタートアッププログラムに追加してみます。

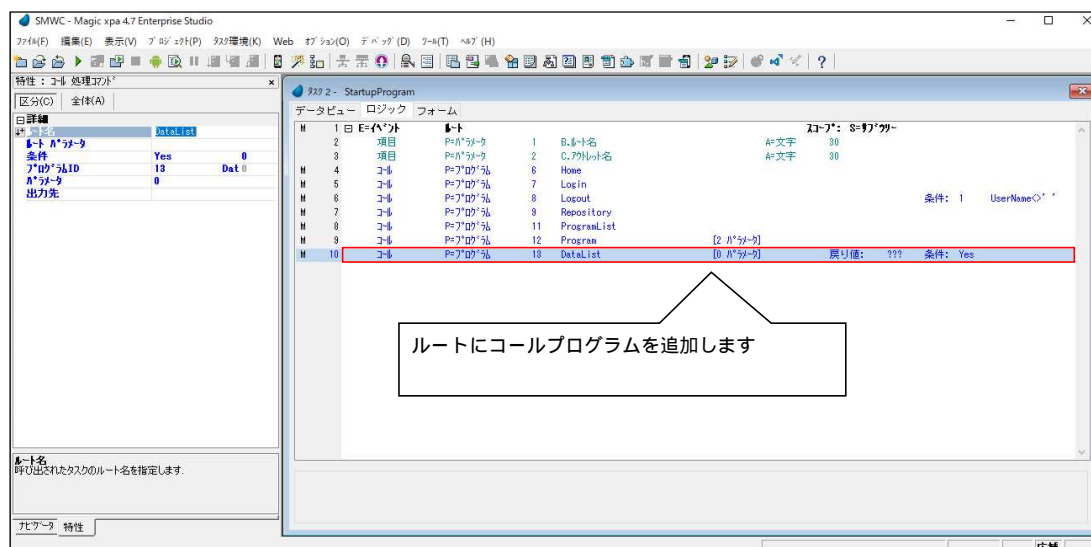


図2.3.2-5 MAGICでのスタートアッププログラム修正

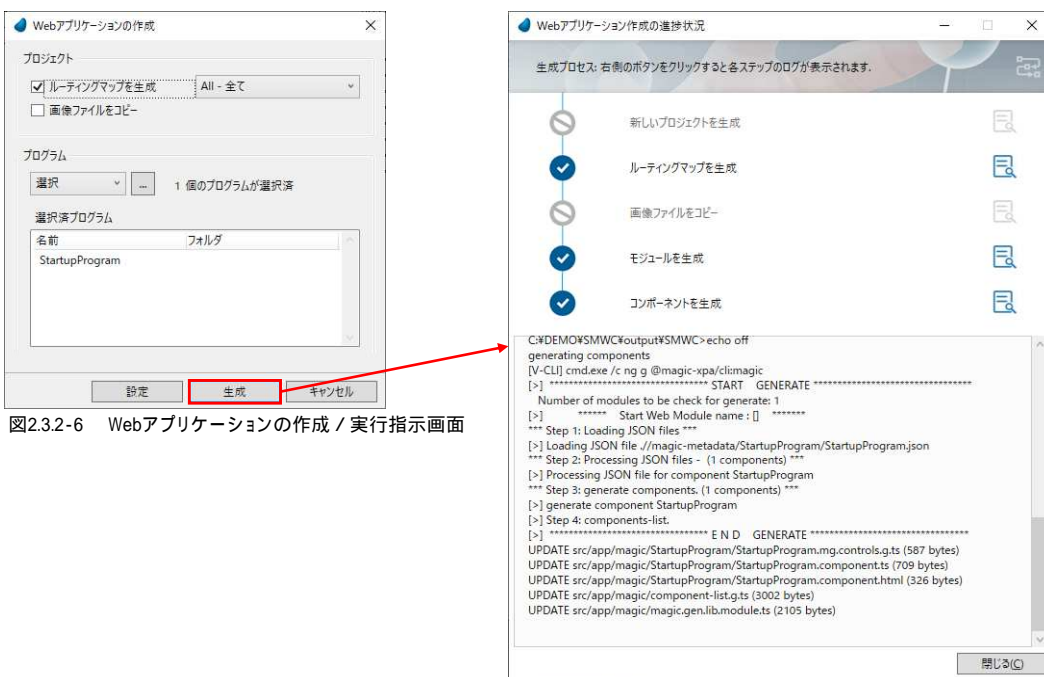


図2.3.2-6 Webアプリケーションの作成 / 実行指示画面

図2.3.2-6 Webアプリケーションの作成 / 進捗表示画面

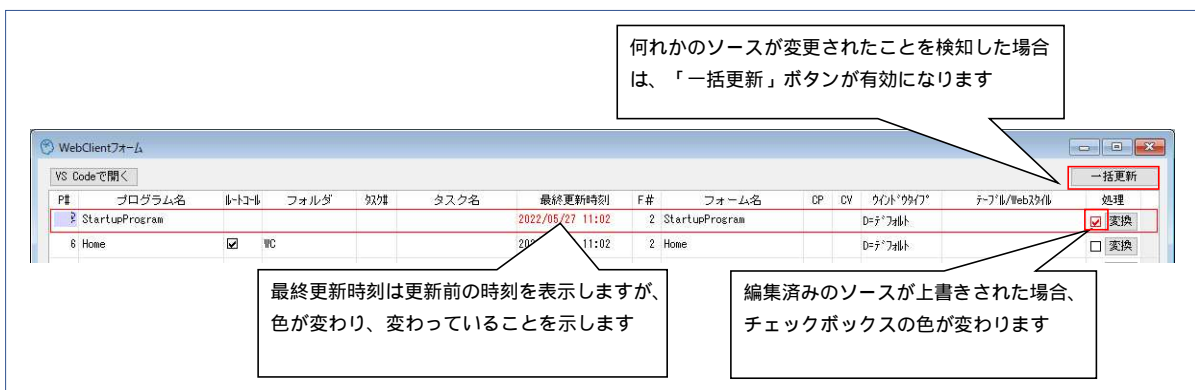


図2.3.2-7 WebClientフォーム画面

「一括更新」ボタンを押すと下記のメッセージを表示します。

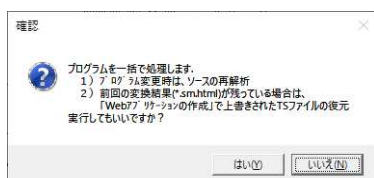


図2.3.2-8 WebClientフォーム / 「一括更新」ボタン押下時のメッセージ

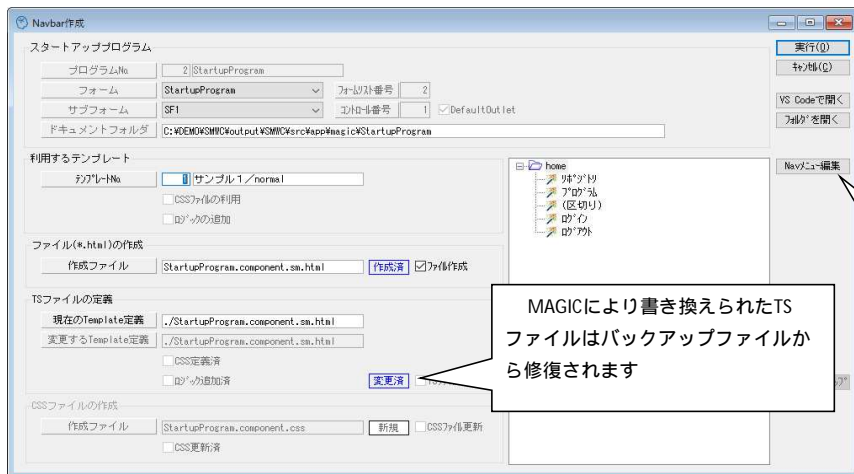


図2.3.2-9 Navbar作成 (WebClientフォーム / スタートアッププログラムでの「変換」ボタン押下)



図2.3.2-10 ルート選択 (メニュー定義 / 「ルート情報取得」ボタン押下)

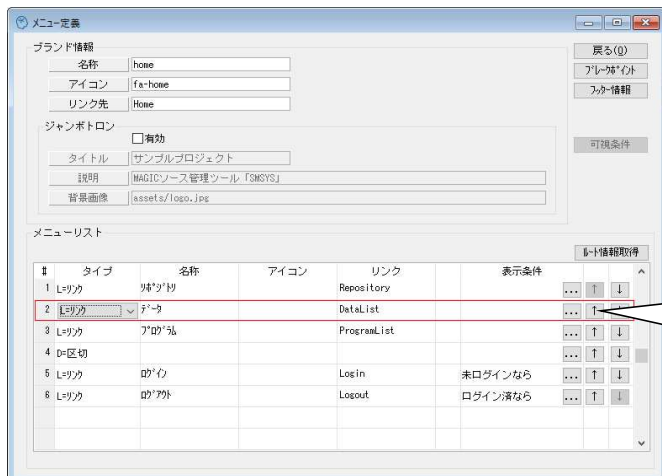


図2.3.2-11 修正が完了したメニュー定義画面

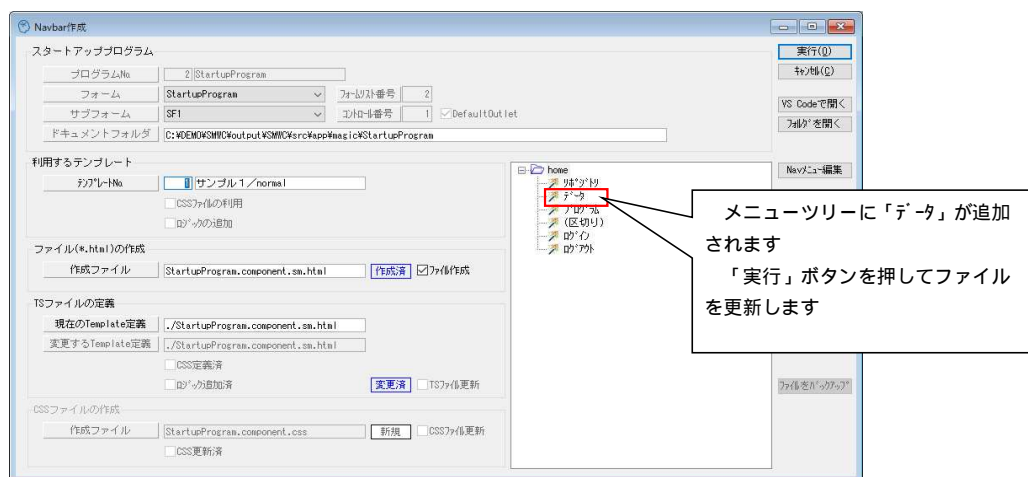


図2.3.2-12 メニュー編集が完了したNavbar作成画面

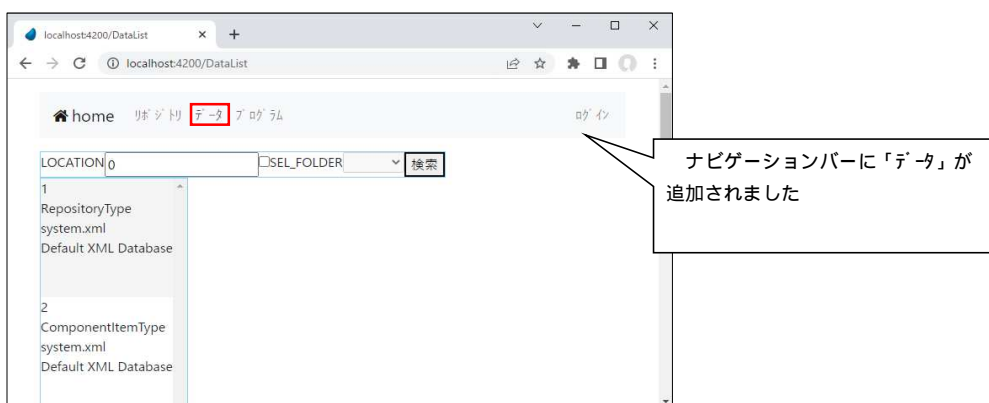


図2.3.2-13 ナビゲーションバーに追加されたメニュー

2.3.3 WebClientフォーム

全プロジェクト内のWebClientフォームの一覧を表示します。

MAGICの「Webアプリケーションの作成」によるAngularソースの更新を自動検知します。これにより、変更が不要なタスクについては一括でリカバリ等の処理が可能

P#	プログラム名	ルートコール	フォルダ	タスク名	最終更新時刻	F#	フォーム名	CP	CV	ウインドウタイプ	テーマ/Webスタイル	処理
1	StartupProgram				2022/05/27 11:02	2	StartupProgram			D=デフォルト		<input checked="" type="checkbox"/> 変換
6	Home	<input checked="" type="checkbox"/>	WC		2022/05/27 11:02	2	Home			D=デフォルト		<input type="checkbox"/> 変換
7	Login	<input checked="" type="checkbox"/>	WC		2022/05/27 11:03	2	Login			D=デフォルト		<input checked="" type="checkbox"/> 変換
8	Logout	<input checked="" type="checkbox"/>	WC		2022/05/27 11:03	2	Logout			D=デフォルト		<input type="checkbox"/> 変換
9	Repository	<input checked="" type="checkbox"/>	WC		2022/05/27 11:03	2	Repository		...	D=デフォルト	R=繰り返し可能な項目	<input checked="" type="checkbox"/> 変換
10	LoginOverlay	<input type="checkbox"/>	WC		2022/05/27 11:03	2	LoginOverlay			D=オーバーレイ		<input type="checkbox"/> 変換
11	ProgramList	<input checked="" type="checkbox"/>	WC		2022/05/27 11:03	2	ProgramList			D=デフォルト		<input type="checkbox"/> 変換
11	ProgramList		WC	.1	2022/05/11 18:20	3	ProgramTbl			D=デフォルト	R=繰り返し可能な項目	<input type="checkbox"/> 変換
12	Program	<input checked="" type="checkbox"/>	WC		2022/05/27 11:04	2	Program			D=デフォルト		<input type="checkbox"/> 変換
12	Program		WC	.1	2022/05/27 11:03	3	ProgramTask			D=デフォルト		<input type="checkbox"/> 変換
12	Program		WC	.1.1	2022/05/27 11:03	4	DataView			D=デフォルト	R=繰り返し可能な項目	<input type="checkbox"/> 変換
12	Program		WC	.1.2	2022/05/27 11:03	4	LogicView			D=デフォルト	R=繰り返し可能な項目	<input type="checkbox"/> 変換
12	Program		WC	.1.3	2022/05/27 11:04	4	FormList			D=デフォルト	R=繰り返し可能な項目	<input type="checkbox"/> 変換
13	DataList	<input type="checkbox"/>	WC		2022/06/06 18:28	2	ProgramList			D=デフォルト		<input type="checkbox"/> 変換
13	DataList		WC	.1	2022/06/06 18:28	3	DataTbl			D=デフォルト	R=繰り返し可能な項目	<input type="checkbox"/> 変換

図2.3.3-1 WebClientフォーム

WebClientフォーム		
項目	説	明
テーブル	プロジェクト内の全WebClientフォームを表示します。	
P#	プログラム番号を表示します。	
プログラム名	プログラム名を表示します。	
ルートコール	スタートアッププログラムのルートでコールされているかどうかを表示します。	
フォルダ	フォルダ名を表示します。	
タスク#	タスクの階層を表す番号を表示します。	
タスク名	タスク名を表示します。	
最終更新時刻	タスクの最終更新時刻を表示します。	
F#	フォーム番号を表示します。	
フォーム名	フォーム名を表示します。	
CP	カスタムプロパティの有無を表示します。表示される「...」ボタンで明細を確認できます。 「図2.3.3-2」	
CV	カスタムバリデータの有無を表示します。表示される「...」ボタンで明細を確認できます。	
ウインドウタイプ	ウインドウタイプ(D=デフォルト / O=オーバーレイ)を表示します。	
テーブル / Webスタイル	テーブルが存在するとき、Webスタイル(R=繰り返し可能な項目 / M=マテリアルデザイングリッド / H=HTMLテーブル)を表示します。	
処理	Angularソースの編集(テンプレートの変換)を行う場合は「変換」ボタンを押します。 (但し、スタートアッププログラムに設定しているプログラムの場合は「Navbar作成」を起動します。「図2.3.2-2」) 編集したデータがある場合はチェックボックスにチェックが入ります。 変換処理の詳細については「2.4 テンプレートの変換」を参照してください。	
「VS Codeで開く」ボタン	Visual Source CodeでAngularソース(現在のTSファイルで指定されたtemplateUrl)を開きます。	
「一括更新」ボタン	差異がある場合に一括で更新を行います。差異とは次のものです。 新しくWebClientタイプのプログラムが作成された場合 プログラムが編集されたため解析済のデータよりもプログラム編集時刻が新しくなった場合 Webアプリケーションの作成が実行され、前回テンプレート更新時のTSファイルが書き変わった場合 の場合は、ソースの再解析、の場合はTSファイルの復元処理を実行します。	

名前	C#	エディタ名	式#	式
count	6	name	2	Trim(Str(C, '52'))

図2.3.3-2 カスタムプロパティ画面

2.3.4 プログラム一覧

全プロジェクト内のプログラム一覧を表示します。
プログラムソースの更新を自動検知します。

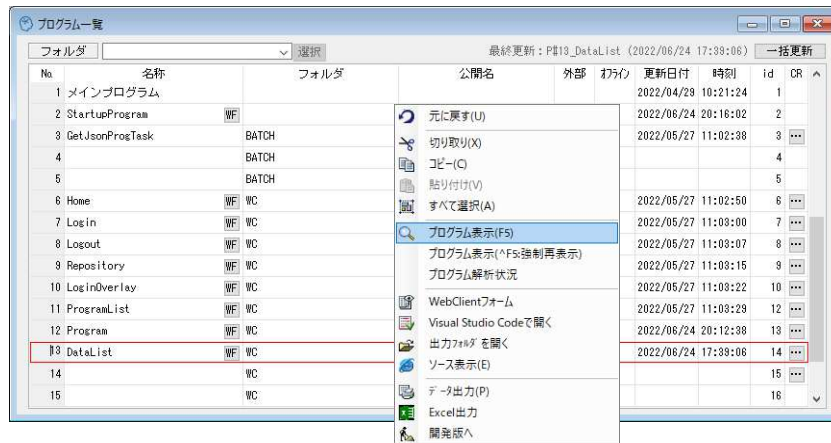


図2.3.4-1 プログラム一覧とコンテキストメニュー

プログラム一覧		
項目	説	明
フォルダ	コンボボックスからフォルダを選択し、「選択」ボタンを押すと絞り込みを行うことが可能です。	
最終更新	最後に更新されたプログラムの情報を表示します。	
「一括更新」ボタン	解析処理後に変更されたプログラムを再度一括して解析します。	
テーブル	プロジェクト内の全プログラムを表示します。	
	プログラム番号を表示します。	
名称	プログラム名を表示します。WebClientタスクの場合は「WF」ボタンを表示します。	
フォルダ	ボタンを押すと、そのプログラムのWebClientフォームを表示します。【図2.3.4-2】	
公開名	フォルダ名を表示します。	
外部	公開名を表示します。	
オフライン	外部属性を表示します。	
更新日付	オフライン属性を表示します。	
時刻	プログラムの更新時刻(日付)を表示します。	
id	プログラムの更新時刻(時刻)を表示します。	
CR	プログラムのID Nを表示します。	
	プログラムが他のプログラムからコールされているときに「...」ボタンを表示します。	
	プッシュボタンを押すとコール元の位置を表示します。	
コンテキストメニュー		
プログラム表示	解析処理後、プログラム内容を表示します。オプションの「更新プログラムの自動解析」にチェックしている場合は、常に新しい状態に解析処理を行います。	
プログラム解析状況	そうでない場合は、「プログラム表示(F5:強制再表示)」で更新できます。	
WebClientフォーム	プログラムの解析状況を表示します。	
Visual Studio Codeで開く	WebClientフォームの一覧を表示します。	
出力フォルダを開く	Visual Source CodeでAngularソース(現在のTSファイルで指定されたtemplateUrl)を開きます。	
ソース表示	Angularソースフォルダをエクスプローラーで開きます。	
開発版へ	プログラムソースファイル(ファイル名:「Prg.###.xml」)をブラウザで表示します。	
	MAGIC xpa Studioを開き、該当プログラムに位置付けします。	

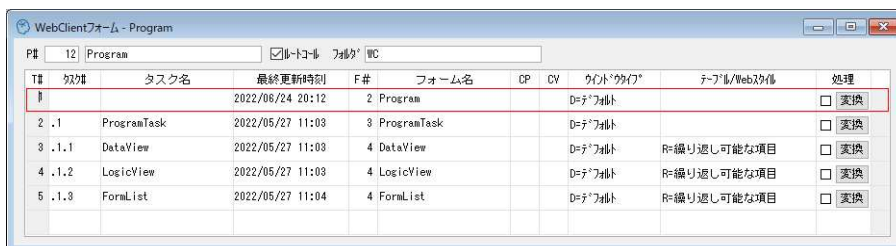


図2.3.4-2 プログラム一覧 / WebClientフォーム

2.3.5 ブルダウンメニュー

ブルダウンメニューに配置されたプログラム中、主要なものについて説明します。

ファイル(F) 編集(E) オプション(O) ヘルプ(H)

ブルダウンメニューから起動する主要なプログラム

項目	説明
ファイル(F)	ファイル関連のメニュー (図2.3.5-1)
登録済みプロジェクトの起動	登録済みのプロジェクトを別インスタンスで起動することが可能です。 (図2.3.5-2)
フォーム状態の初期化	ほぼすべてのプログラムは変更したウィンドウのサイズ、テーブルのカラム幅等を記憶します。これらを初期状態に戻す場合は、このメニューを使用して下さい。
MAGIC特殊パラメータ	MAGIC.INIの特殊パラメータとその現在の値を表示します。値を変更したい場合は、「修正」モードに変更します。
アプリケーションの再起動	現在のシステムを再起動します。再起動後も現在表示しているプロジェクトを読み込みます。
オプション(O)	オプション関連のメニュー (図2.3.5-3)
TABで移動	「TAB」キーで項目を移動するかどうかを指定します。テキストのコピー等が不要な場合はオフにします。初期値は「環境設定」プログラムで決めることが可能です。
更新プログラムの自動解析	プログラム一覧等で、ズームしたときにその内容を表示しますが、プログラムの更新があった時、無条件に再解析をおこなうかどうかを指定します。初期値は「環境設定」プログラムで決めることが可能です。WebClientフォームの「設定」ボタンでテンプレートの変換プログラムを起動しようとした場合は、この設定に関係なく更新を検知した場合は無条件に更新(自動解析)します。
ヘルプ(H)	ヘルプ関連のメニュー (図2.3.5-4)
環境設定	動作条件の設定以外に、データ定義 (図2.3.5-5) やフォント・表示色 (図2.3.5-6) の設定が可能です。WebClientではないプロジェクトの場合は、このプログラムでデータ定義を行ってください。
登録プロジェクト一覧	登録してあるプロジェクトを一覧で確認したり定義内容を編集するための保守用プログラムです。また、最近のプロジェクトや既存のデータベースから登録するための機能を用意しています。 (図2.3.5-7)
Web上のヘルプ	Webサイトからヘルプページを開きます (準備中)
DB設定ウィザード	2.2.5 / ① 参照 (初期設定アシスタントから呼び出す同名のプログラムと同じ)
Bootstrap設定ウィザード	2.2.5 / ② 参照 (初期設定アシスタントから呼び出す同名のプログラムと同じ)
Angular環境設定	Bootstrap設定ウィザードで自動設定を行うためのデータを定義します。 (図2.3.5-8)
ライセンス登録	評価用ライセンスではメニューは非表示になります。ライセンスファイルを読み込み、登録を行います。 (図2.3.5-9)
バージョン情報	バージョンを表示します。 (図2.3.5-10)

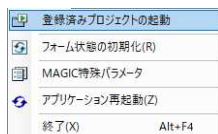


図2.3.5-1 「ファイル」ブルダウンメニュー

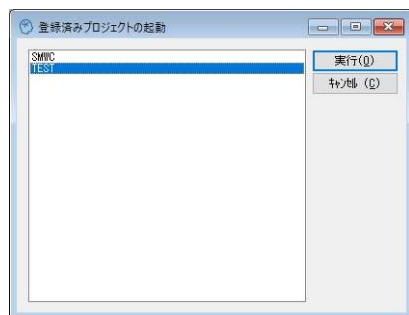


図2.3.5-2 登録済みプロジェクトの起動



図2.3.5-3 「オプション」ブルダウンメニュー



図2.3.5-4 「ヘルプ」プルダウンメニュー

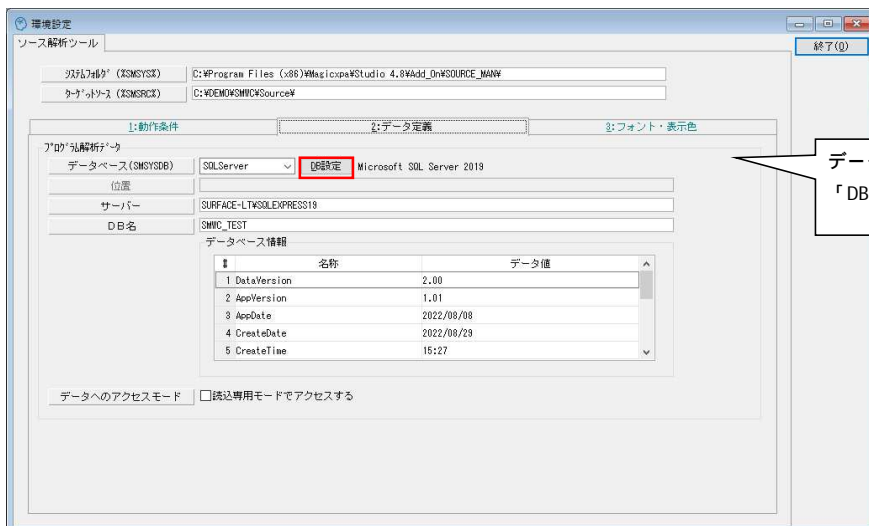


図2.3.5-5 環境設定 / データ定義

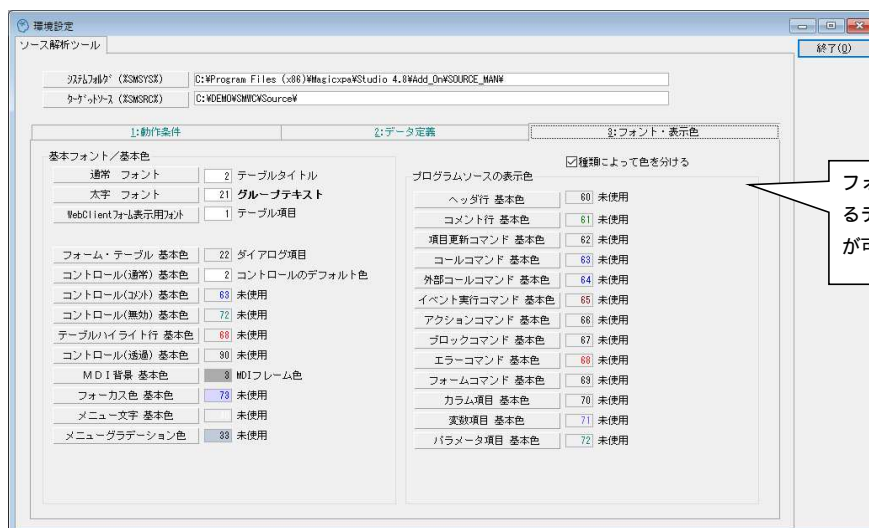


図2.3.5-6 環境設定 / フォント・表示色

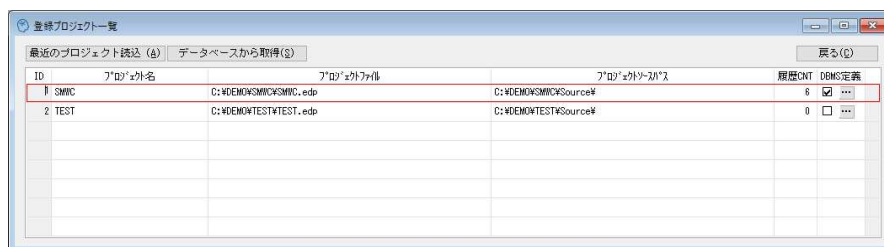


図2.3.5-7 登録プロジェクト一覧

登録プロジェクト一覧は、登録したプロジェクトの保守用のプログラムです。
設定したデータベースとの不一致が発生した場合には、「DBMS定義」のチェックを外すことによりエラーを回避することが可能です。
また、EDPファイルを指定する以外の方法でプロジェクトを登録することが可能です。

様々なプロジェクト追加方法

項目	説明
最近のプロジェクト読込	開発版で開いたことのあるプロジェクトであればその履歴情報からプロジェクトを選択できます。開くことができるMAGICのバージョンに制限はありません。コンボボックスからMAGICのバージョンを選択すると、それにマッチするプロジェクトの一覧を表示します。
データベースからの取得	SMSYSで定義したデータベース (Microsoft SQL Serverに限定) であれば、選択するだけでプロジェクトの追加が可能です。表示されるコンボボックスからSQL Serverを選択し、更にデータベースを選択して下さい。

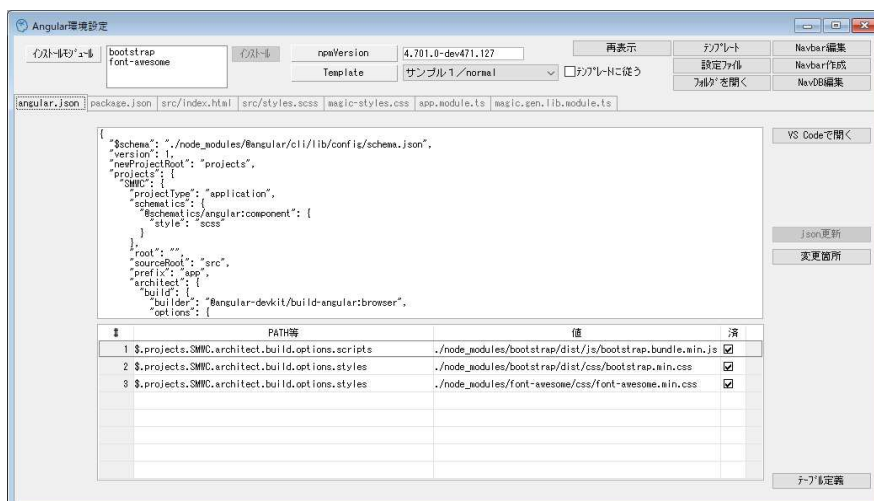


図2.3.5-8 Angular環境設定

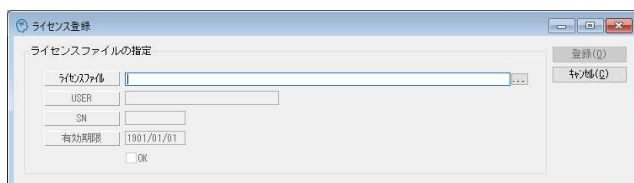


図2.3.5-9 ライセンス登録



図2.3.5-10 バージョン情報

2.4 テンプレートの変換

2.4.1 概要

「テンプレートの変換」プログラムは、「楽々ウェブクライアント」のもっとも中心的な機能です。
MAGICが出力したAngularソースのデザイン情報を変換します。

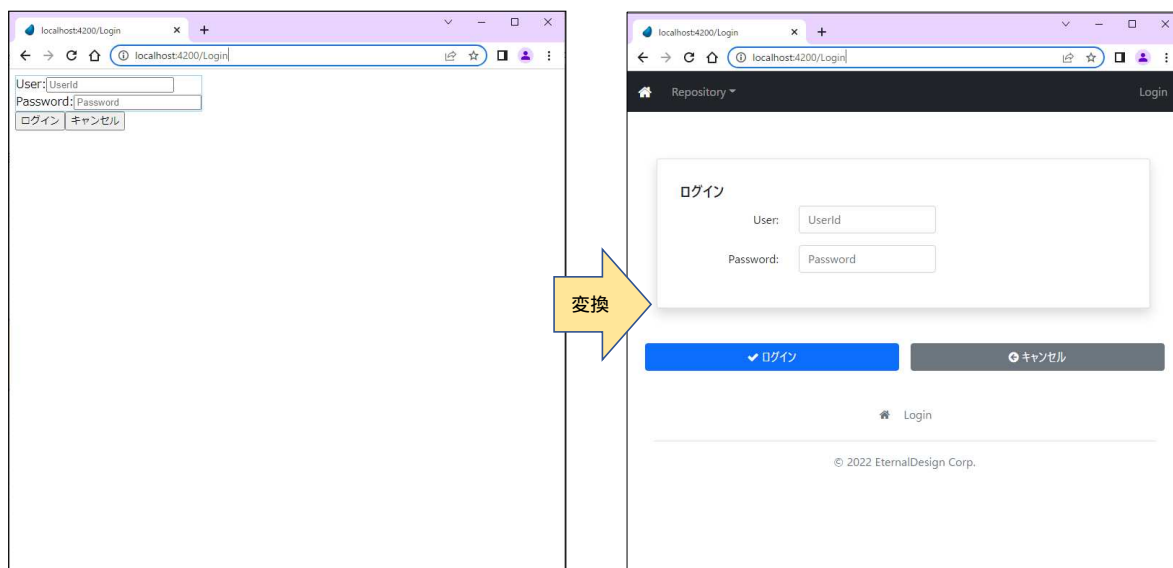


図2.4.1-1 「テンプレート変換」によるWebデザインの変更例

特長

予め設定した変換ルールにより、テンプレートファイル(TSファイルのtemplateUrlで設定)をダイナミックに変換することが可能です。
コントロール別にBootstrapのクラス情報を設定することが可能です。生成は一括して自動的なので、コーディングレスな編集が可能です。
カスタムプロパティで設定した値を利用して、要素値の表示や可視、属性値に設定することが可能です。
ダイアログでの変換を指示するだけで、テンプレートファイルの作成他、CSSファイルの生成やTSファイルの書き換え、バックアップ等を自動的に行います。
編集情報はデータに残るため、Angularソースが全て失われたとしても再生成させることが可能です。

変換ルール

MAGICが出力したテンプレートファイルのHTMLをタグ単位でダイナミックに変更するためのもので、下記の処理の組み合わせを予めセットしたものです。

要素名を書き替え
子要素を作成 / 親要素を作成
属性の追加 / 削除
属性値の追加 / 変更
リボジトリ情報からテーブルヘッダ部を生成

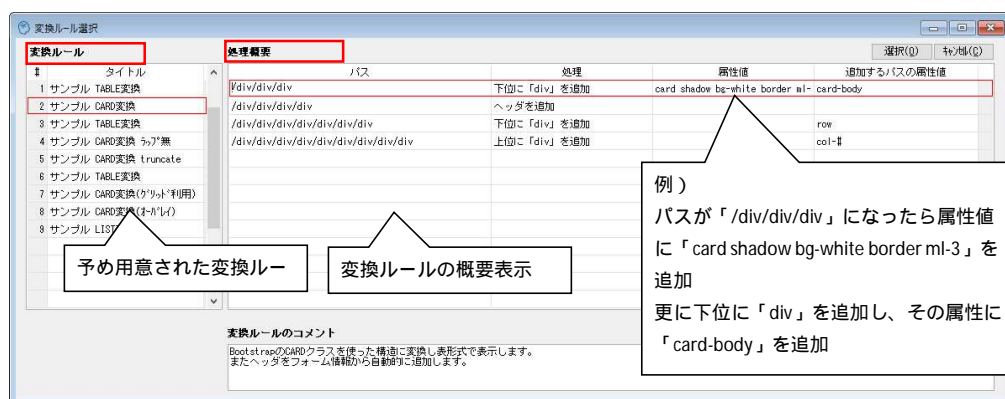


図2.4.1-2 変換ルールの選択画面

コントロール属性

コントロール別にダイアログを表示して、Bootstrapのクラスを設定することが可能です。

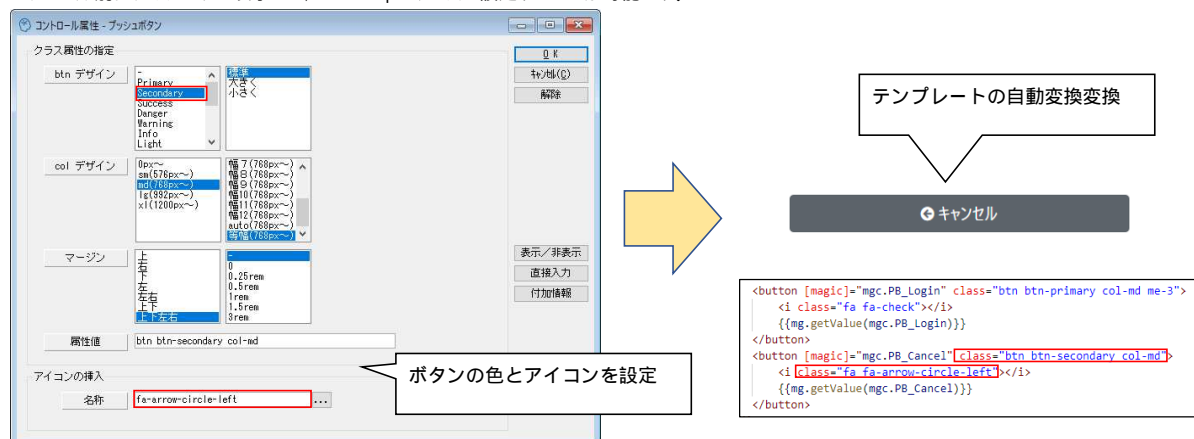


図2.4.1-3 コントロール属性 (プッシュボタン)

カスタムプロパティの利用

カスタムプロパティを使った情報を、コントロール属性の「付加情報」から参照して入力に利用することが可能です。

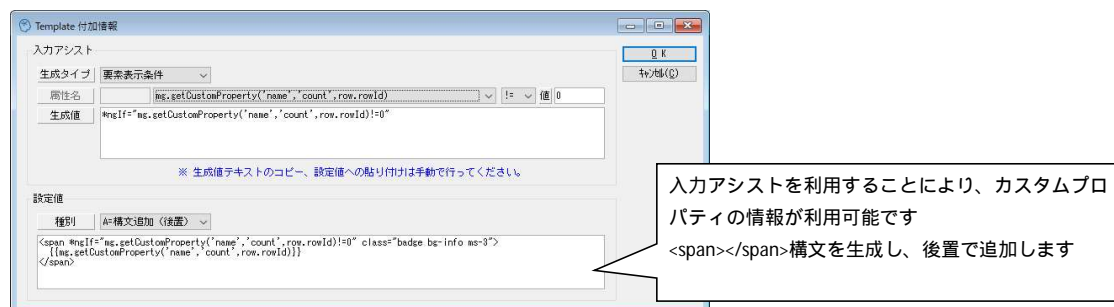


図2.4.1-4 コントロール属性 / 付加情報

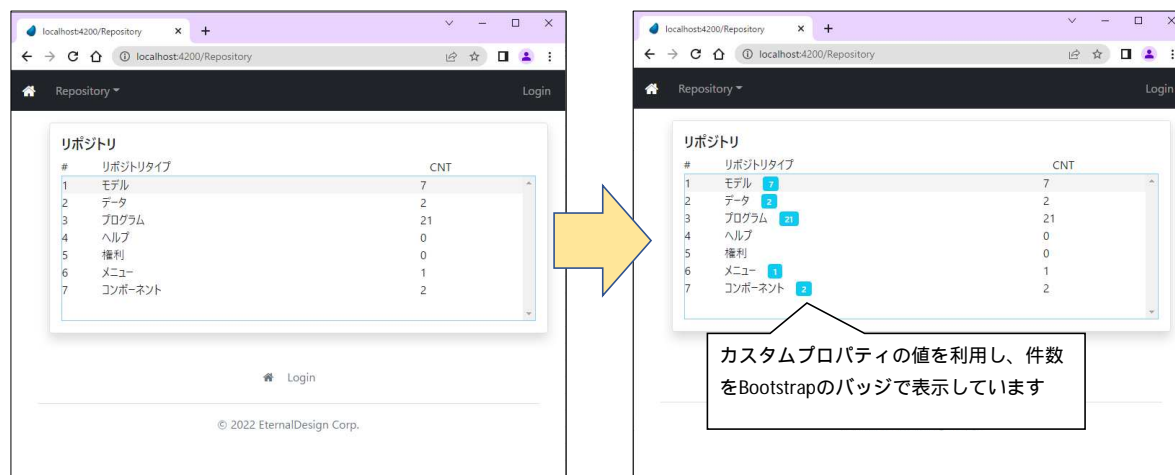


図2.4.1-5 カスタムプロパティの利用例

ダイアログによる変換指示画面

「テンプレートの変換」プログラムのダイアログを下图に表示します。
htmlファイルの変換と共に、関連するファイルとその更新が確認可能な構造になっています。

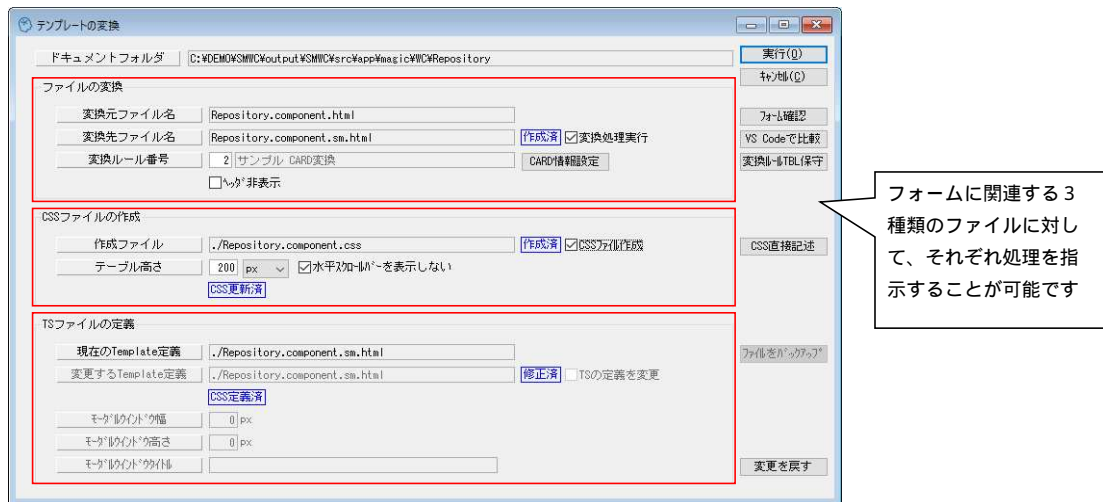


図2.4.1-7 テンプレートの変換

処理の概要を下图に示します。
テンプレート変換のダイアログで入力した情報、コントロール属性はデータベースに格納されます。
これにより、Angularソースは何らかの不測の事態が起きたとしてもいつでも復旧が可能になります。

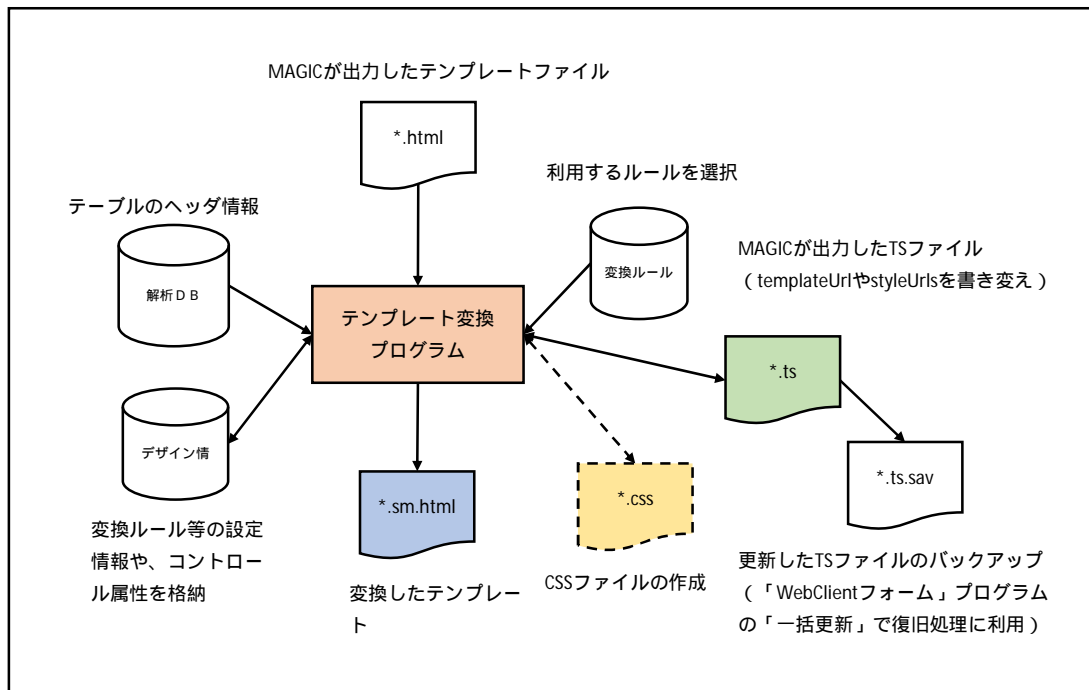


図2.4.1-8 テンプレート変換プログラムの処理

2.4.2 テーブルの変換

1) 変換ルールの利用

変換ルールによりテーブルを処理する例についてサンプルプロジェクトを例に説明を進めます。

WebClientフォームを起動します。

先ず、テーブルのWebスタイルが「R=繰り返し可能な項目」に設定されていることを確認し、「変換」ボタンを押します。



図2.4.2-1 WebClientフォーム

テンプレートの変換プログラムを起動したら、「変換処理実行」をチェックします。

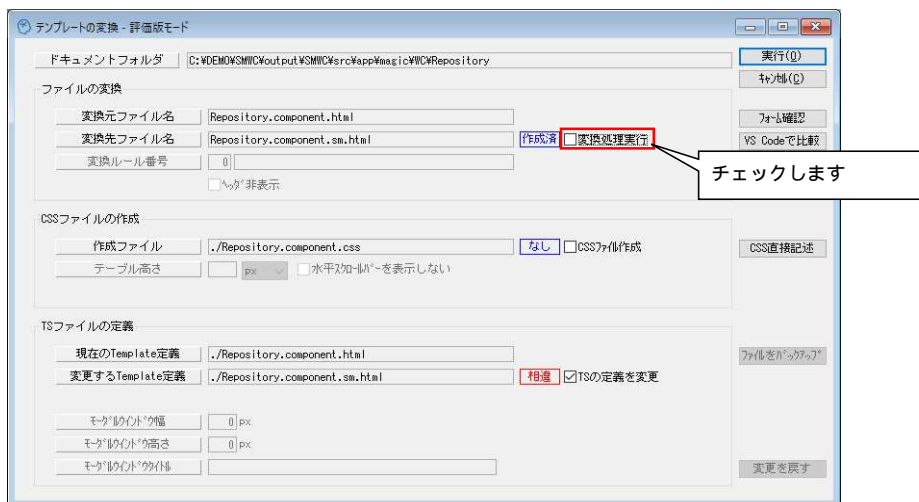


図2.4.2-2 WebClientフォーム / テンプレートの変換 (起動直後)

変換ルール番号を入力します。

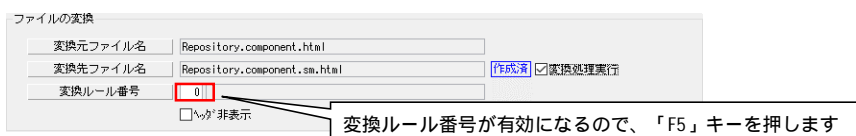


図2.4.2-3 変換ルール番号

1番の「サンプル TABLE変換」を選択します。

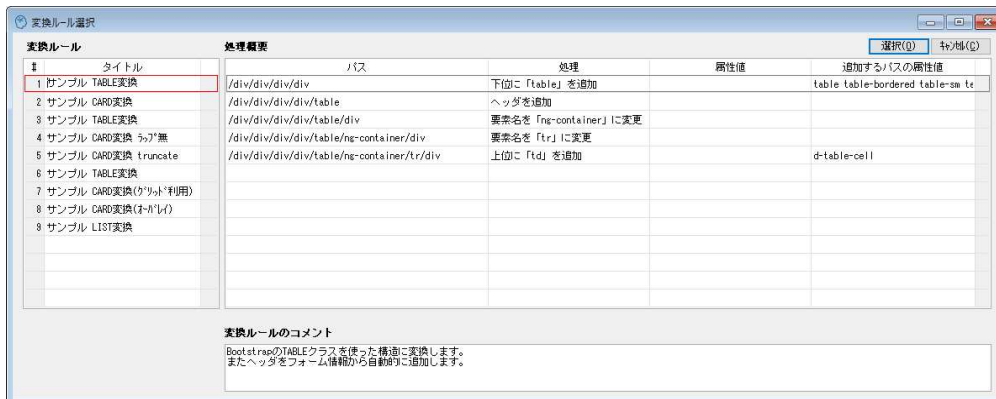


図2.4.2-4 変換ルール選択

変換の準備ができました。「実行」ボタンを押して下さい。

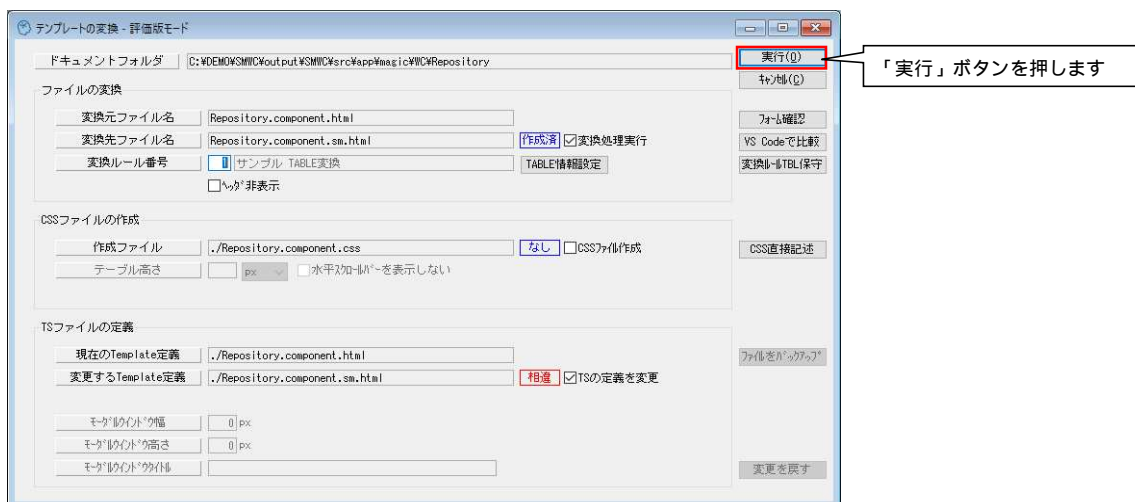


図2.4.2-5 テンプレート変換(変換ルール指定後)

開発版をプロジェクト実行し、VS Codeのターミナル画面でサーバを起動('ng serve'コマンド)します。
ブラウザ画面のURLに'http://localhost:4200/Repository'を入力します。

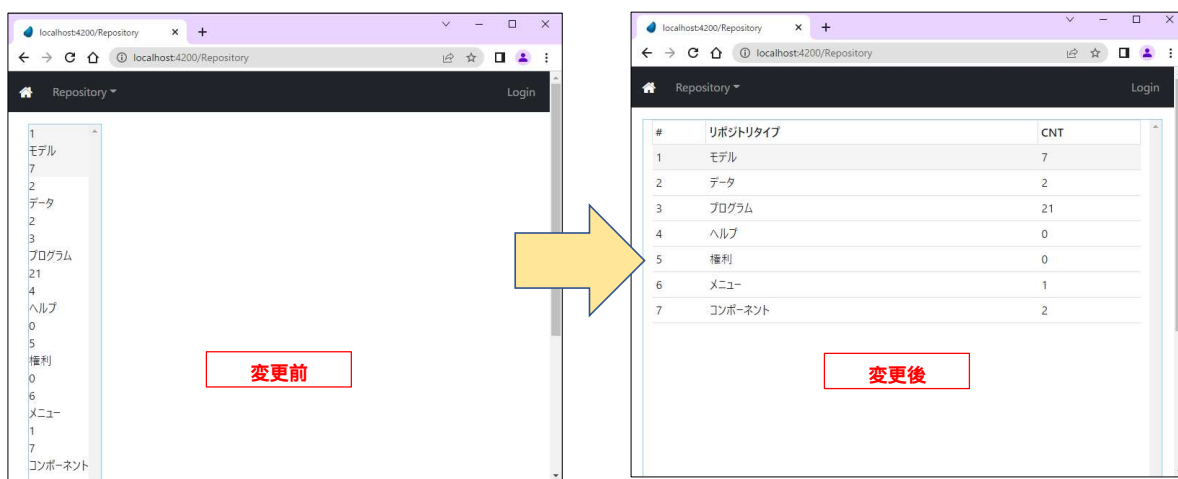


図2.4.2-6 変換ルールを適用したWeb画面 (TABLE変換)

CARD変換

再び変換画面を表示します。
今度は、変換ルール番号に2番の「サンプル CARD変換」を選択します。

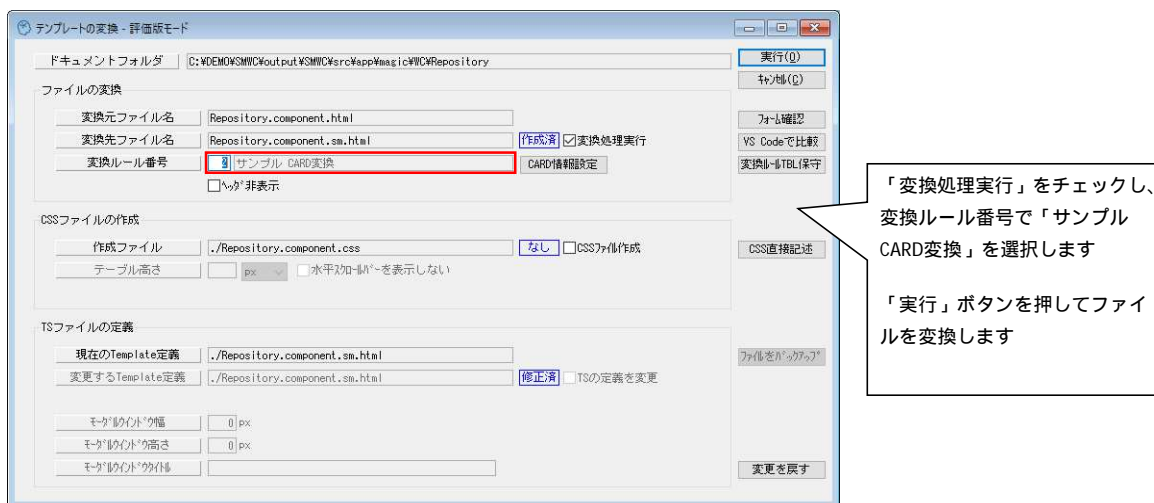


図2.4.2-7 CARDタイプの変換ルールを指示

作成されたCSSファイルは「作成ファイル」のファイル名のコンテキストメニューからすぐに確認が可能です。

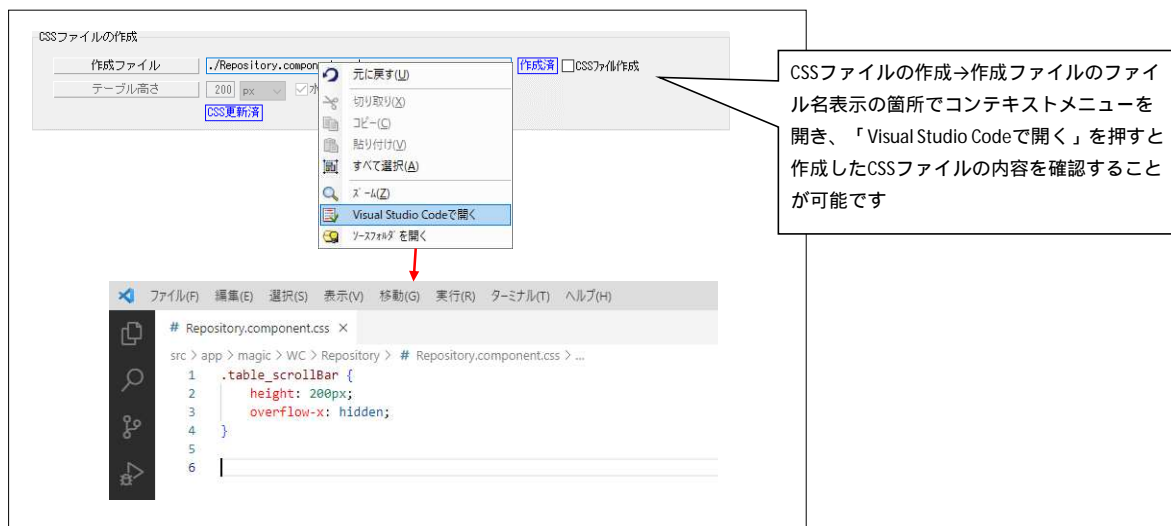


図2.4.2-11 作成したCSSファイルと内容確認方法

3) ヘッダ処理

ここで、楽々ウェブクライアントのテーブルヘッダの処理について説明します。

自動生成の場合

MAGICのWebアプリケーションの作成において、テーブルコントロールのWebスタイルが「R=繰り返し可能な項目」のときは、ヘッダのタイトルは出力されません。これを補うために、解析したりボジトリ情報を読み込み、自動的に生成しています。

また、カラムの幅についても、開発時のフォーム画面で設定したカラム幅の割合に近くなるよう計算し、サイズを処理しています。

カラム幅の自動生成

タイプ	説明
tableクラス	'<th style="width: ##%">タイトル</th>'のように百分率で幅を指定します。
cardクラス	'<div class="col-#">タイトル</div>'のようにグリッド幅(全体 = 12を案分)で指定します。



図2.4.2-12 「サンプル TABLE変換」で生成されたヘッダ情報

変換ルールに「TABLE変換」を指定したときは、style属性で幅が指定されたヘッダを挿入します
百分率はフォーム上で配置したカラムの割合から算出します



図2.4.2-13 「サンプル CARD変換」で生成されたヘッダ情報

変換ルールに「CARD変換」を指定したときは、Bootstrapのグリッド幅が指定されたヘッダを挿入します
グリッドの幅もカラムサイズから算出しますが、カラム数が多い場合等、合計が12を超えてしまう(=折り返されてしまう)ことがあります

カラム幅を直接指定

変換ルールに「CARD変換」を指定した場合は、グリッド幅を指定することが可能です。
このグリッド幅は、Bootstrapのグリッドシステム(領域を12分割した列で構成)を利用したものです。

自動	1	8	3
変更例	4	3	5

開発画面で編集したフォームのカラム幅を案分

任意のカラム幅

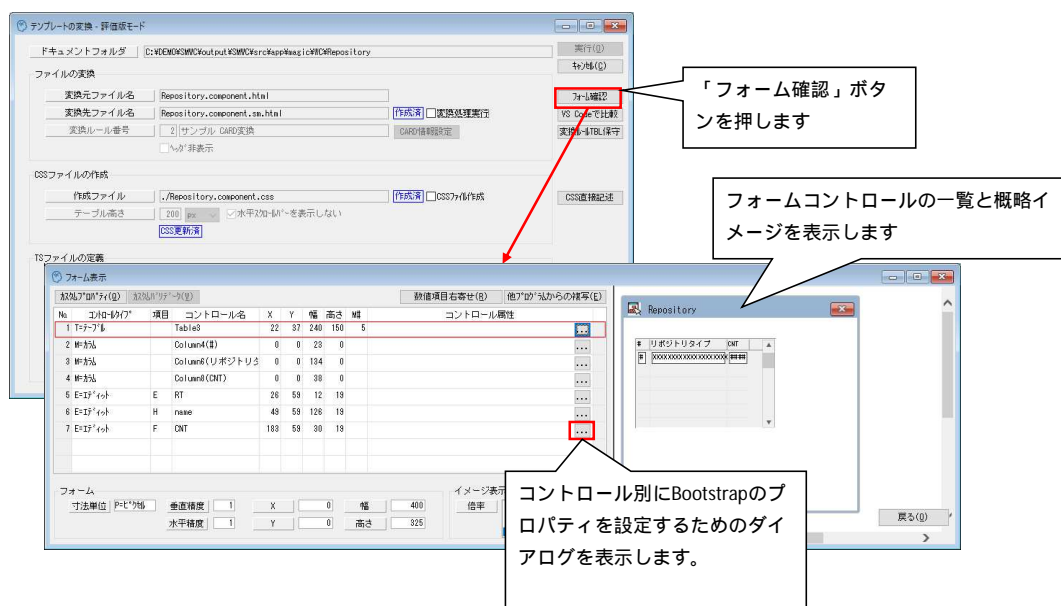


図2.4.2-14 フォーム表示

カラムへの属性割り当て



図2.4.2-15 コントロールの一覧からカラムへの入力

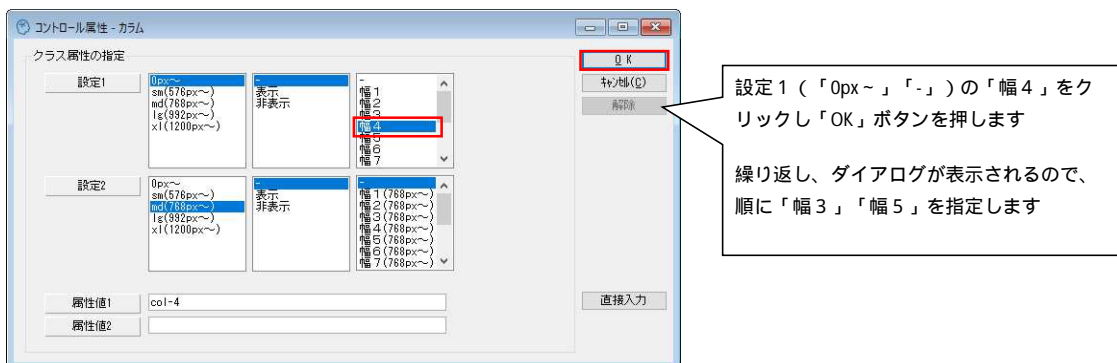


図2.4.2-16 コントロール属性 / カラム設定 (CARDタイプ)

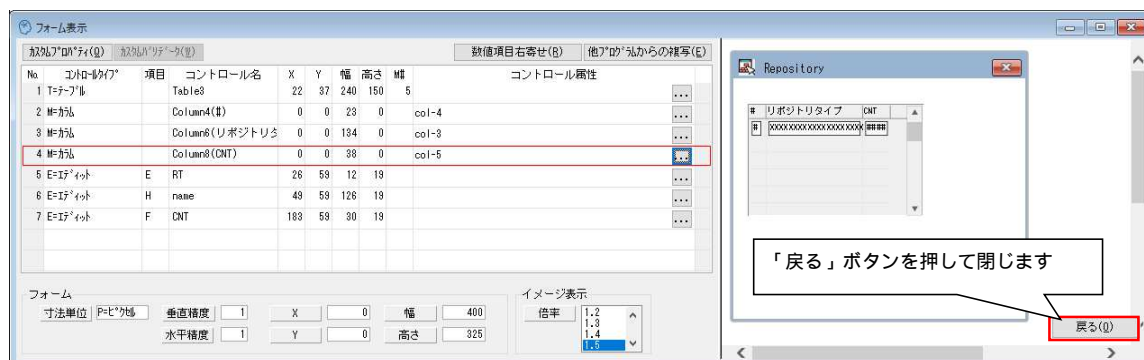


図2.4.2-17 カラム設定終了時の画面

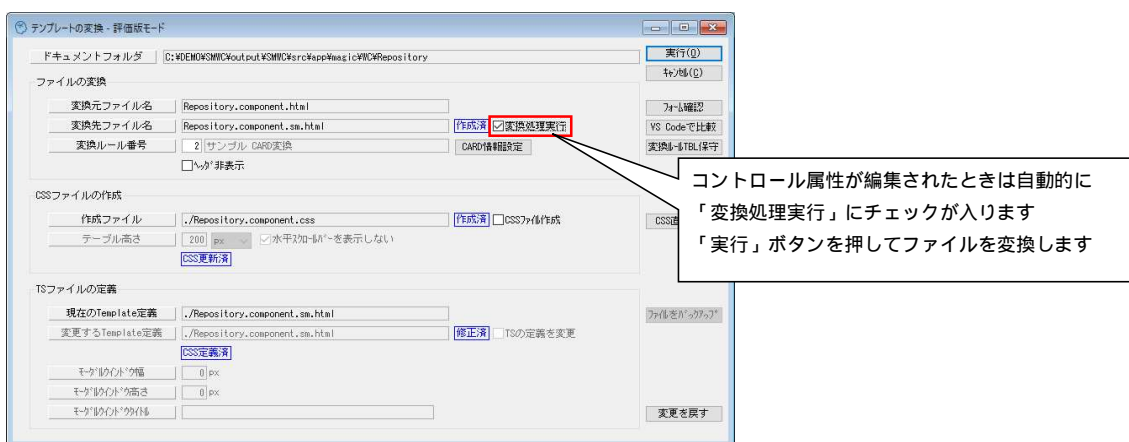


図2.4.2-18 テンプレートの変換(実行直前の画面)

実行画面を確認します。指定した列幅で表示されていることを確認します。

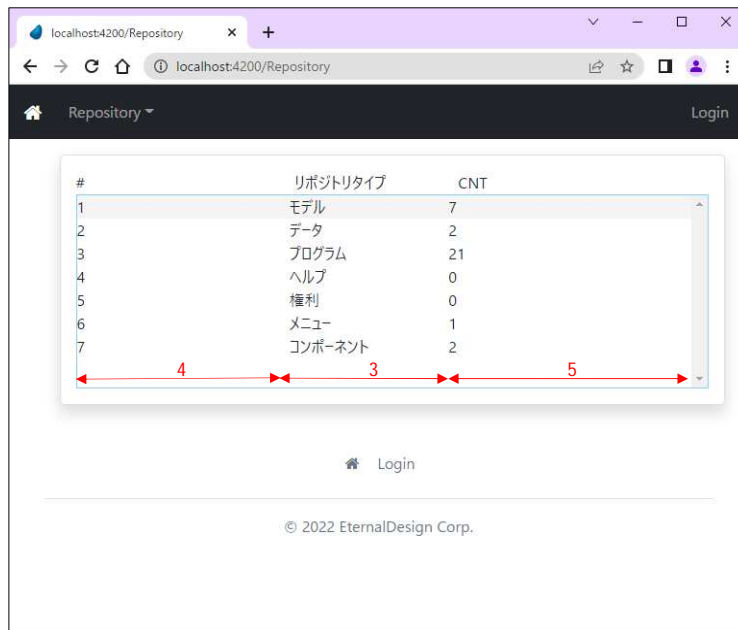


図2.4.2-19 カラム幅をグリッドで指定した場合

カラム幅を非表示にする

カラムの表示 / 非表示を指定することが可能です。

現在	4	3	5
変更	非表示	3	5

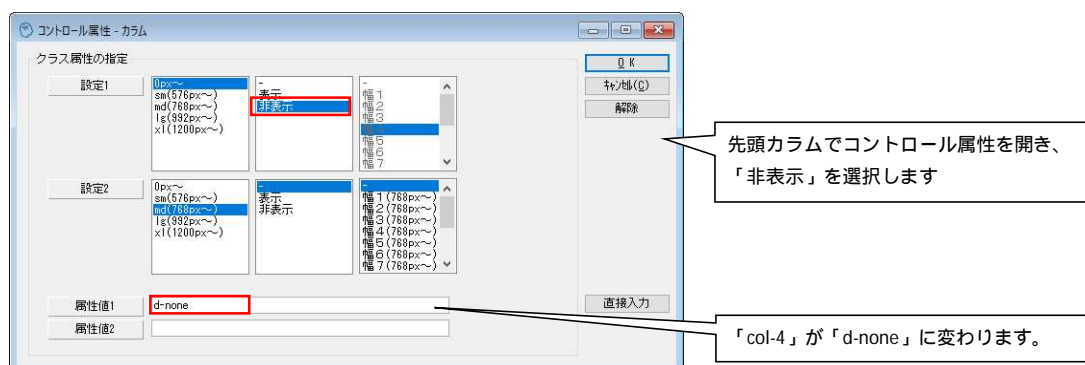


図2.4.2-20 カラムを非表示に設定

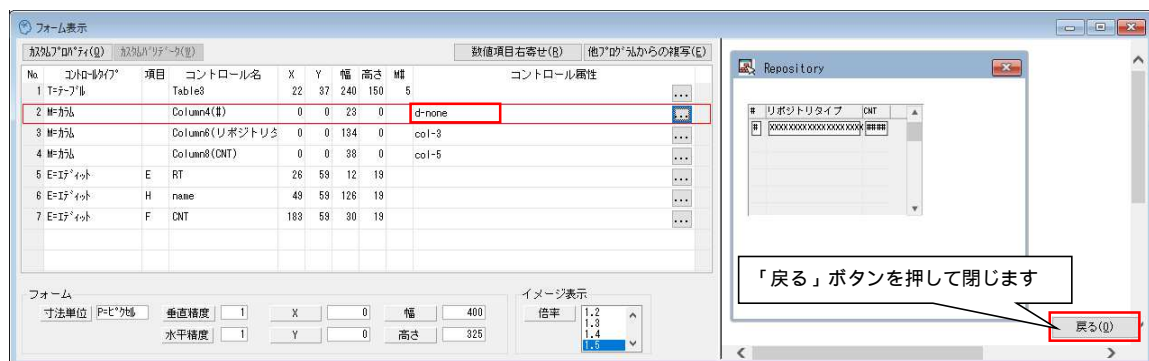


図2.4.2-21 カラム設定を確認

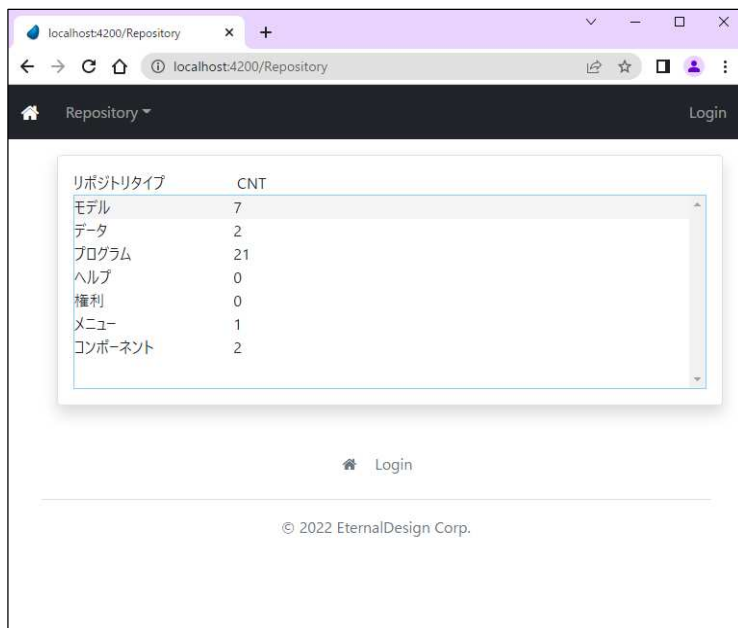
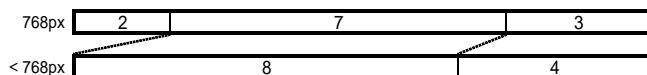


図2.4.2-22 非表示に設定されたカラム

ウィンドウサイズでカラム数を変える

Bootstrapのディスプレイプロパティを使用し、画面サイズに応じて表示 / 非表示を制御することが可能です。

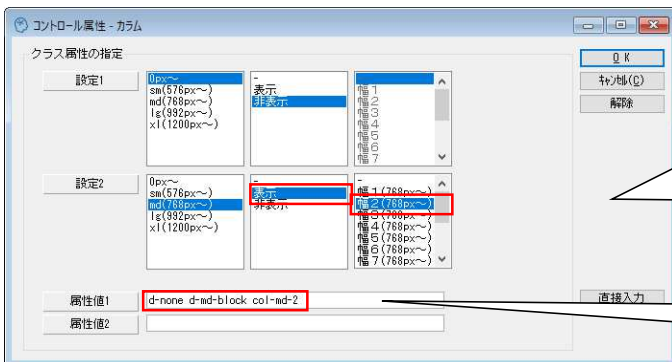
例えば、グリッド幅を下图のように割り当て、サイズが大きい場合は3列に、サイズが小さい時は2列で表示させることが可能です。



画面サイズが768px以上の場合のカラム幅の割合

画面サイズが768px未満の場合のカラム幅の割合

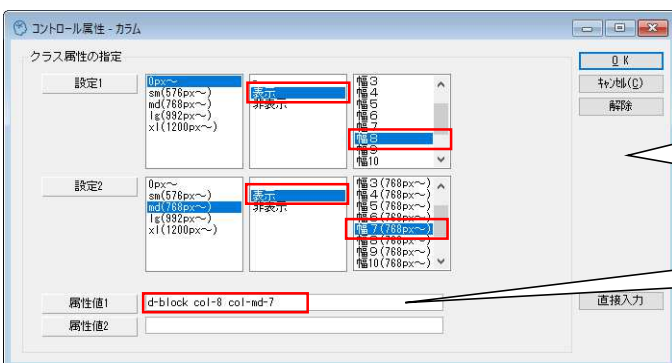
これらの表示制御もコントロール属性の画面から設定が可能です。



先頭カラムでコントロール属性を開き、設定2の「表示」「幅 2 (768px ~)」を選択します。

「d-md-block」と「col-md-2」が追加されます。

図2.4.2-23 先頭カラムの設定



2番目のカラムでコントロール属性を開き、図のように値を設定します。

「d-block col-8 col-md-7」に値が変わります。

図2.4.2-24 2番目のカラムの設定

画面サイズの閾値は、一番左のリストボックスで指定することができます。

3番目のカラムも同様にして設定を変更します。

	画面サイズ	表示 / 非表示	幅
設定1	0px ~	表示	幅 4
設定2	md (768px ~)	表示	幅 3 (768px ~)

設定後の画面は下図のようになります。

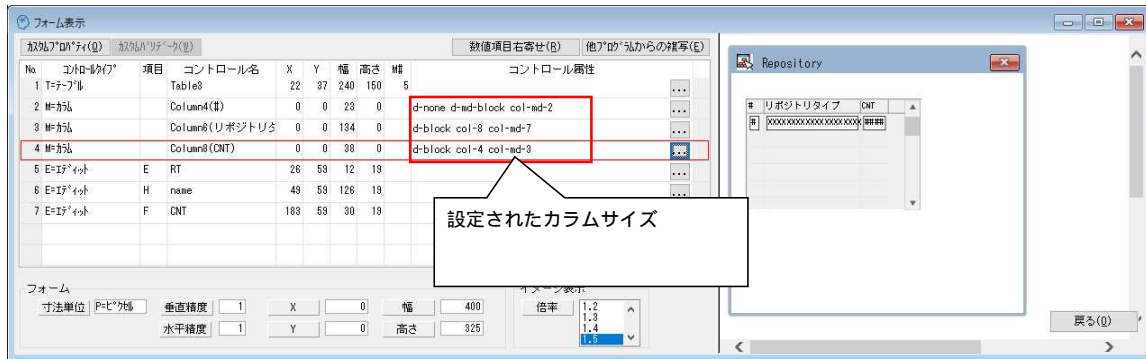


図2.4.2-25 カラムの画面サイズが異なるときの割合を設定

実行結果は下図のようになります。

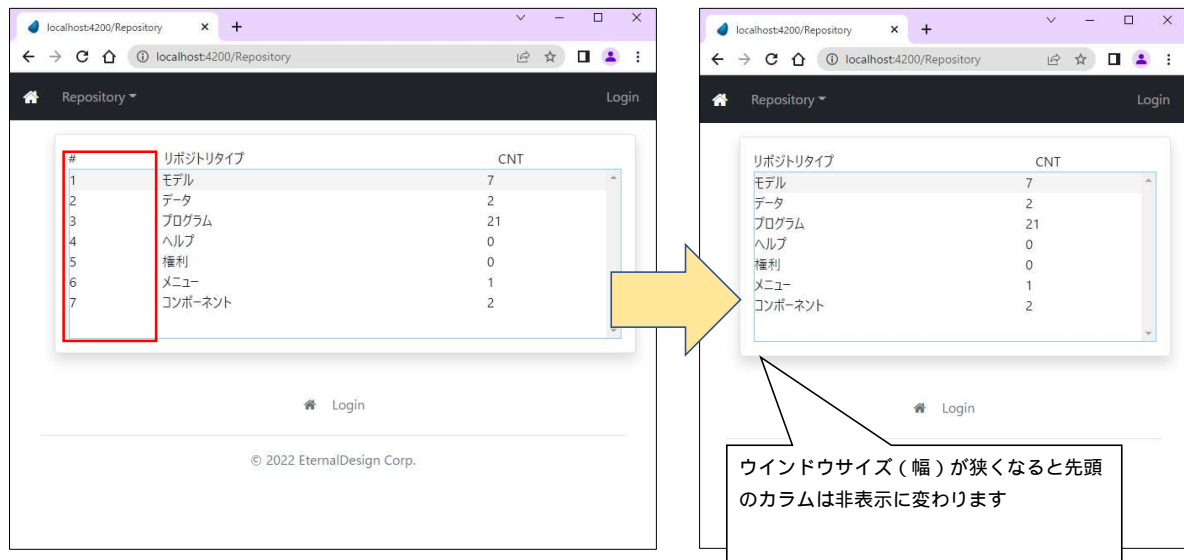
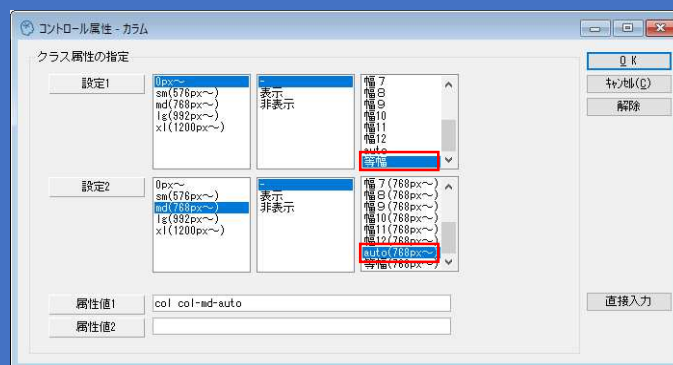


図2.4.2-26 画面サイズが異なるときの表示(カラムの表示 / 非表示)

< 補足 >

bootstrapのカラムでは、「等幅」や「内容に応じた幅」の指定も利用可能です。

当システムでは、幅の指定からそれぞれ「等幅」「auto」を選択します。



4) TABLE / CARDクラス属性の変更

変換ルールTBLで予め設定されたクラス属性の初期値を変更して実行することが可能です。

TABLE属性

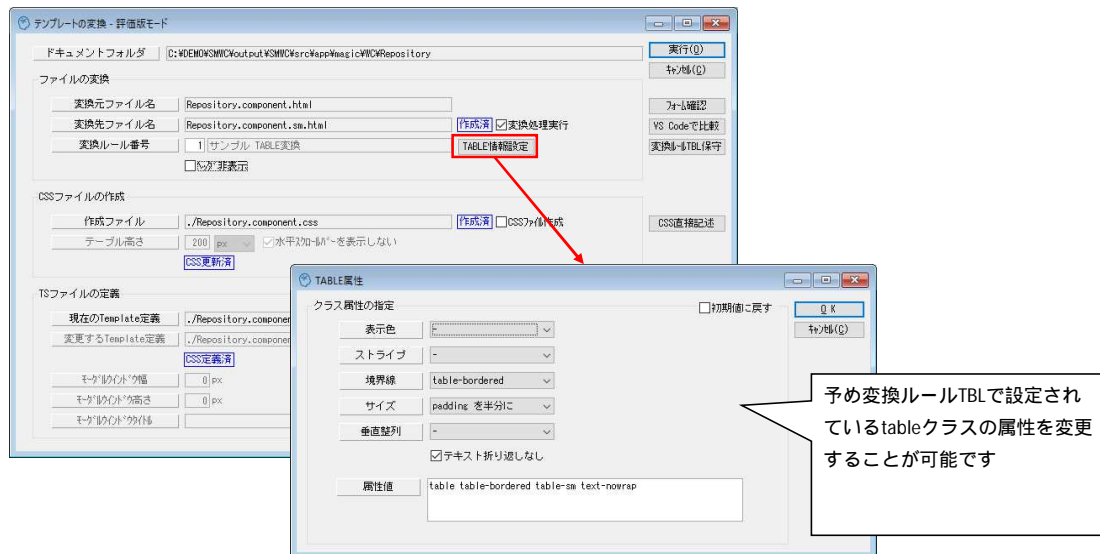


図2.4.2-27 TABLE属性の編集

CARD属性

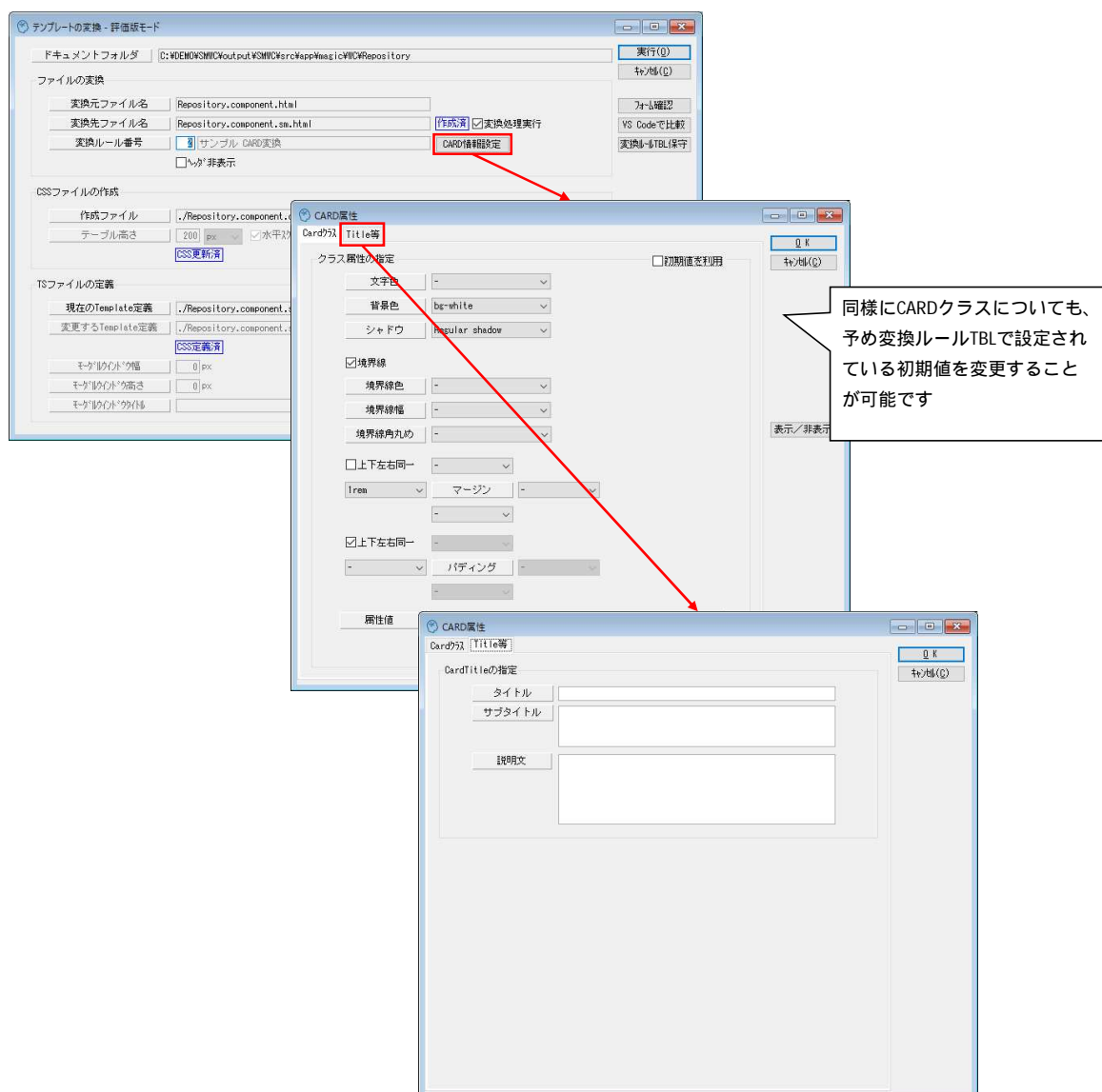


図2.4.2-28 CARD属性の編集

5) 数値項目等の右寄せ処理

テーブルに配置した数値項目を右寄せで表示させるための機能について説明します。
MAGIC xpa WebClientでは、テーブルに配置された項目に関して、処理モードと状態によってHTMLのタグのタイプが入れ替わるようになっています。

テーブルに配置したエディットコントロールの処理について

a) 編集状態

テーブルコントロールが配置されているラインモードのタスクでは、初期モードが「M=修正」の場合に、「行編集に入る」イベントで編集状態になります。
(初期モードが「Q=照会」の場合は「行編集に入る」イベントは受け付けられないので注意)
開発画面では、フォームの位置でAPGを実行し、表示モードを「L=ライン」に指定したときに出現する「行編集ボタンを作成」にチェックすると自動作成が可能

b) 「修正許可」属性

テーブルに配置された数値や文字等のタイプのエディット項目は、「修正許可」属性により、編集可否が決定します。

c) 編集状態の判定と実行されるタグ

状態が編集状態にあるかどうかは関数「mg.isRowInRowEditing」で判定します。(「修正許可」がFALSEの項目は編集状態になりません)
編集状態にある場合 (*ngIf="mg.isRowInRowEditing(row)") は、「input」タグで記載した部分が表示されます。
編集状態にない場合 (*ngIf="!mg.isRowInRowEditing(row)") は、「label」タグで記載した部分が表示されます。
テンプレート上では、1つのエディットコントロールに対して予め上記部分を分けて二重に定義されます。(下図)

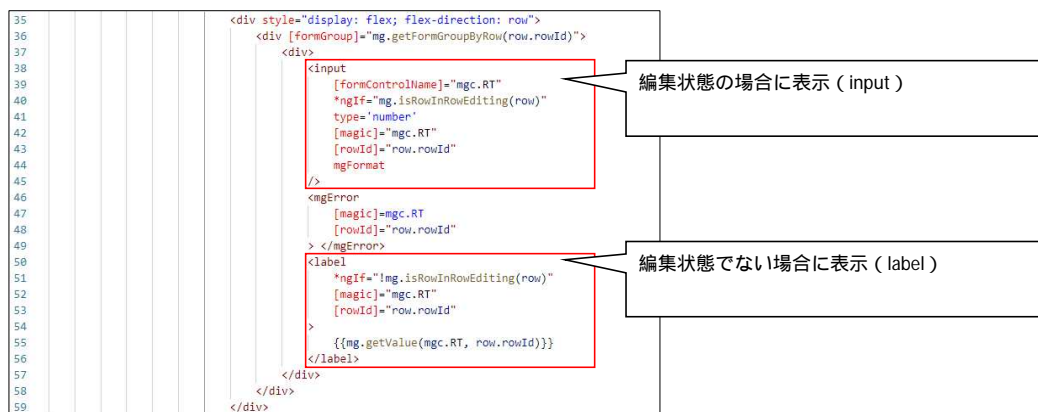


図2.4.2-29 二重に定義されたエディット項目の記載

d) 右寄せ用の方法

「label」タグにclass属性「pull-right」をセットすることにより右寄せが可能です。当機能はこれを自動化するものです。

コントロール属性画面での設定方法

エディットコントロールの属性指定画面を下図に示します。
テキスト処理の「配置」で「右寄せ」にチェックします。

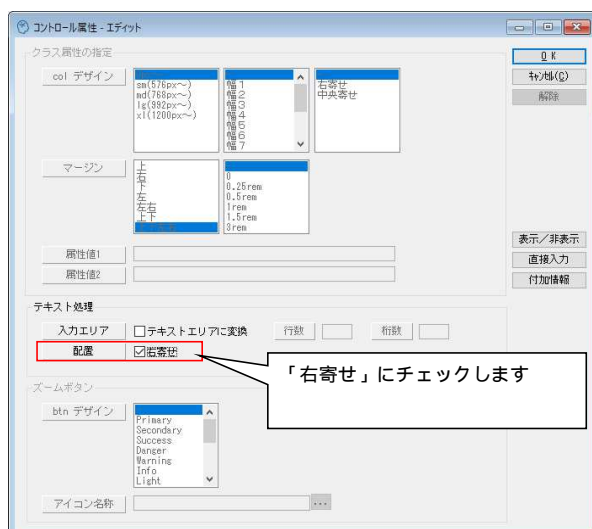


図2.4.2-30 コントロール属性編集によるテキストの右寄せ配置

テーブルに配置されていないエディットコントロール
は対象外です。
予め、無効状態を表示します

「数値項目右寄せ」による一括設定

エディットコントロールのみを一覧で表示させ、一括で右寄せ指定を行うことが可能です。

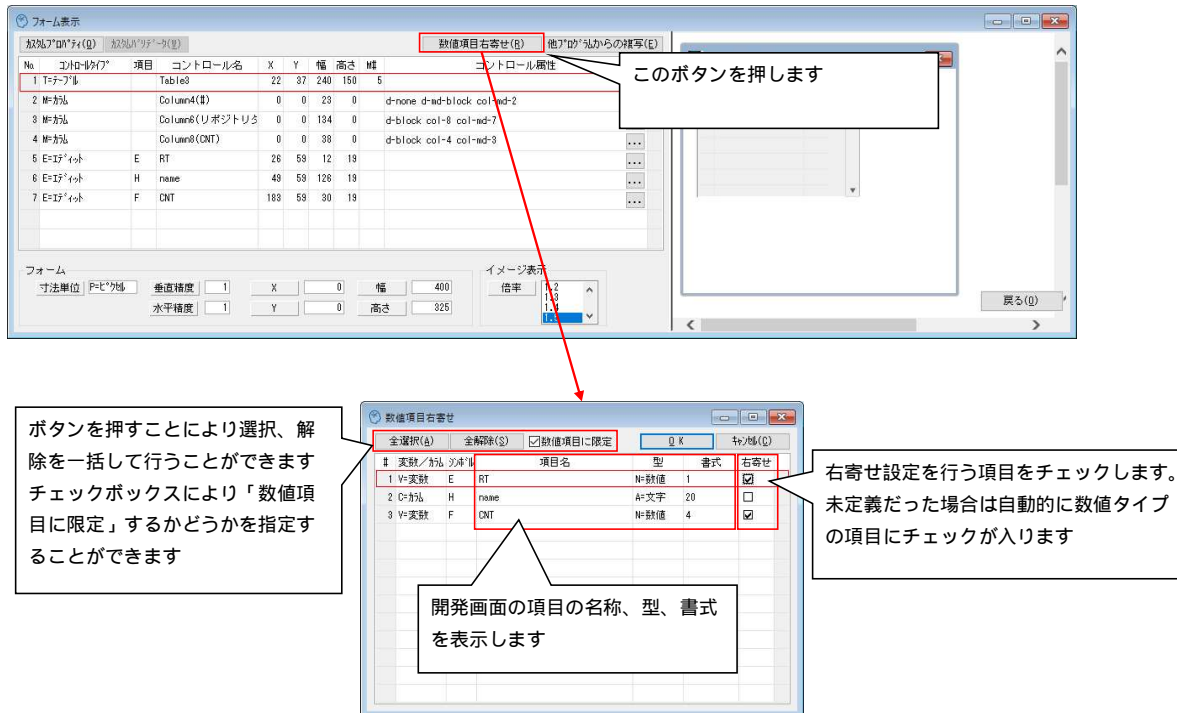


図2.4.2-31 「数値項目右寄せ」画面の呼出し

右寄せ指定した箇所には「pull-right」を表示します。

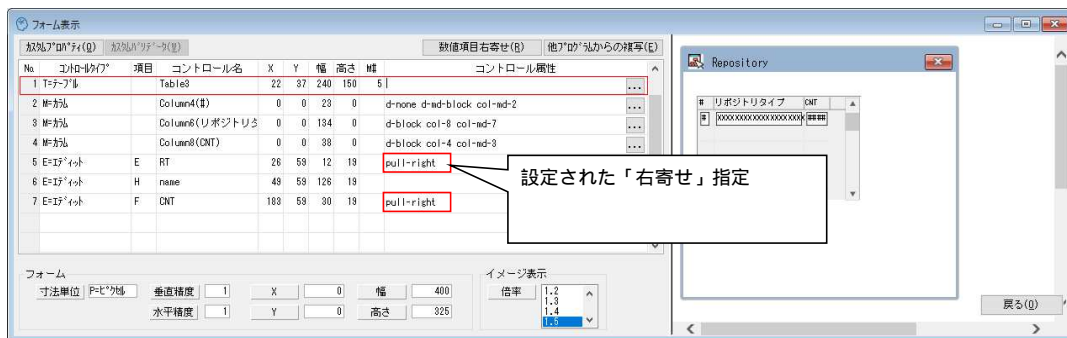


図2.4.2-32 設定された右寄せ指定（フォーム表示）

戻り、テンプレートの変換を実行します。
実行画面で右寄せされていることを確認します。

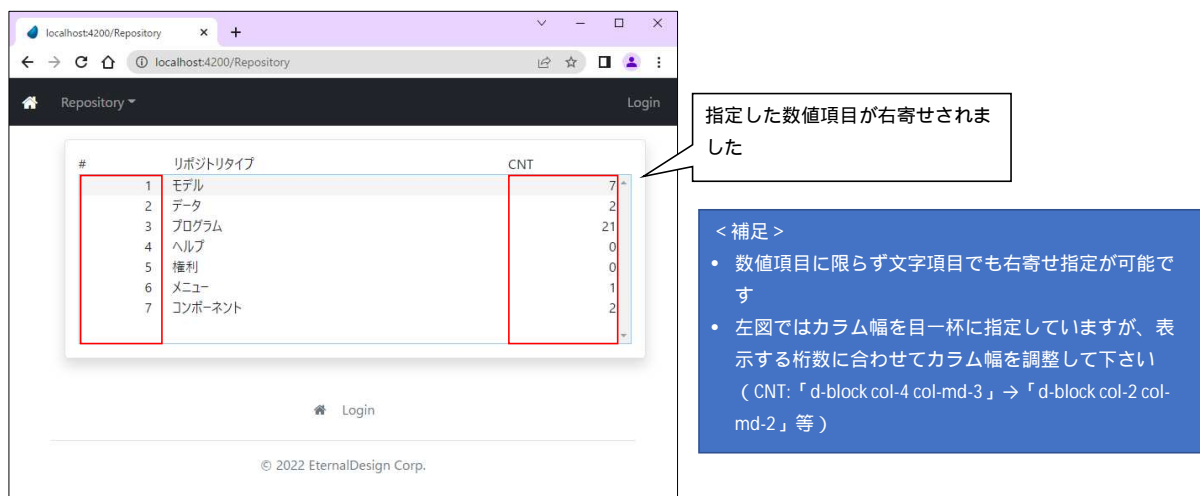


図2.4.2-33 「右寄せ」指定の実行結果

ソースを確認します。

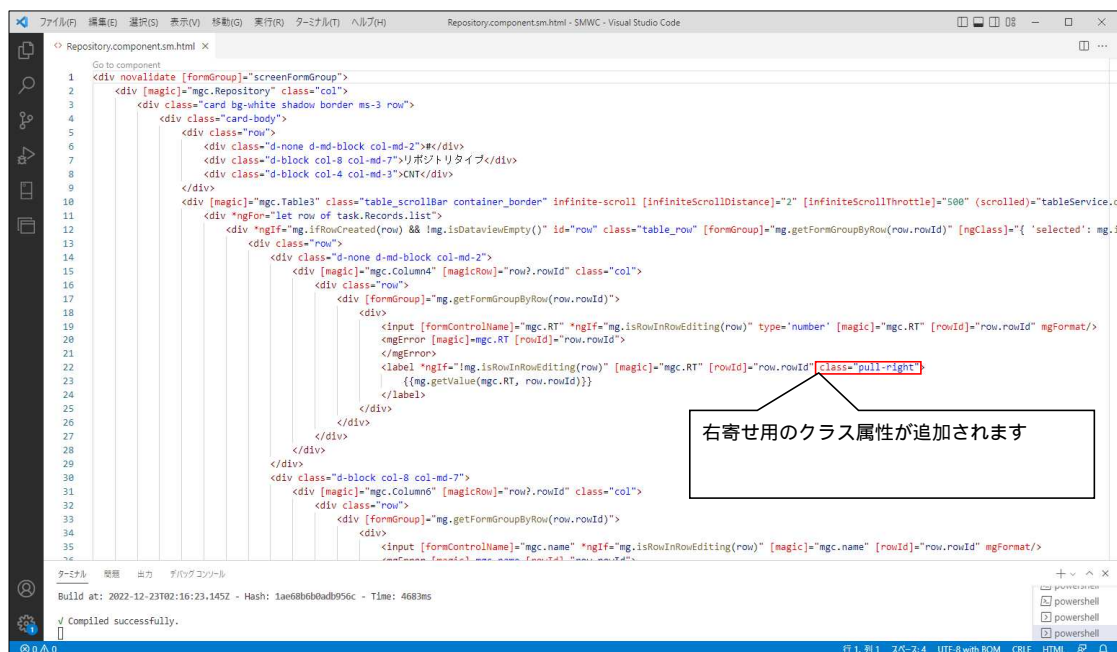


図2.4.2-34 '右寄せ' 指定されたHTML

2.4.3 テーブル定義のないタスクフォームの変換

テーブルを持たないタスクの場合について説明します。

1) 変換ルールの適用範囲

1フォーム内に配置できるテーブルコントロールは1個までです。
それゆえテーブルコントロールを持つフォームの場合は、変換ルールを指定するとテーブルに対して適用します。
つまり、テーブルのあるフォームはテーブルに対する変換ルールを選択することになります。

しかし、フォーム内にテーブルしか配置されていない場合は大きな問題はないですが、例えば、次のようなフォームの場合はどうでしょうか？
もしかしら、テーブル以外の部分 (特に先頭付近の条件を入力するエリア等) に対しては、別な変換ルールを適用させたいと考えるかもしれません。

そこで、当システムでは、上記の画面を下図のようなエリアで分割して、それぞれの部分に対して変換処理を実行できるようにしています。

当システムでの変換ルール割り当て

テーブルがあるときは、変換ルールはテーブルに適用
領域を「/div/div/div[n]」で区分けする
同じ高さに設定したコントロールは同じ領域になる
複数の位置にあるコントロールをまとめたい場合はグループコントロールを配置して関連付ける
領域に対して変換ルールを個別に指定可
テーブル、サブフォーム、タブは除外
個別指定しない場合は、可能な全領域に適用
適用ルールを除外したい領域は個別に指定可能

2) ログインダイアログの変換

ログインダイアログの変換を行ってみます。
このプログラムは、予め開発画面で複数のコントロールをグループ化しています。

図2.4.3-3 Loginプログラムの開発画面

楽々ウェブクライアントのWebClientフォームから#7のプログラム「Login」の変換ボタンを押してテンプレート変換画面を表示します。

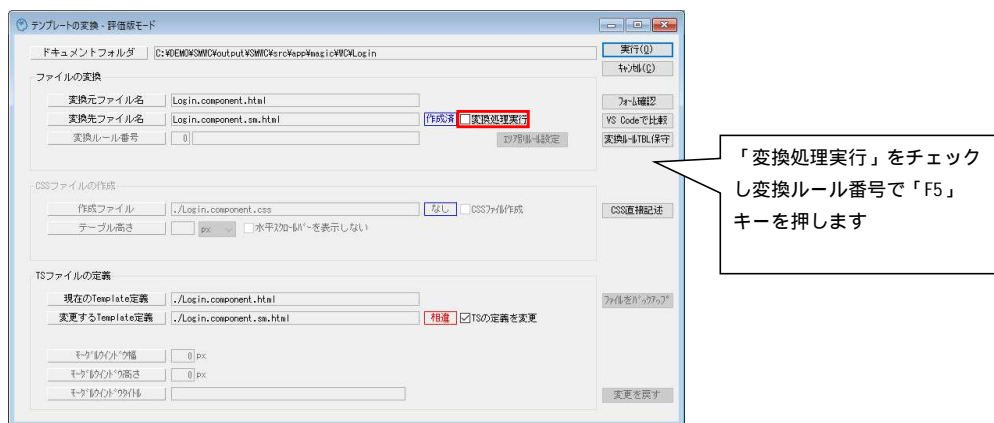


図2.4.3-4 テンプレート変換(Loginプログラム)

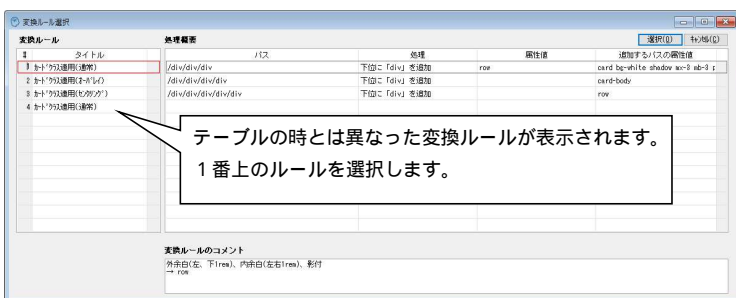


図2.4.3-5 変換ルール選択(テーブル以外の一般ダイアログ用)



図2.4.3-6 変換実行直前の画面

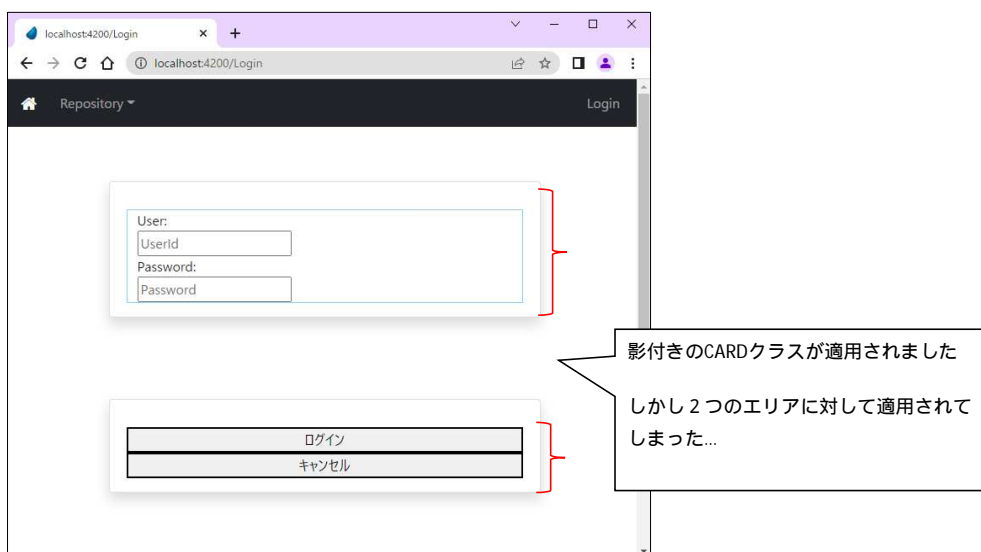


図2.4.3-7 変換後のブラウザ表示

エリア毎にルールを設定するには、「エリア別ルール設定」ボタンを押します。

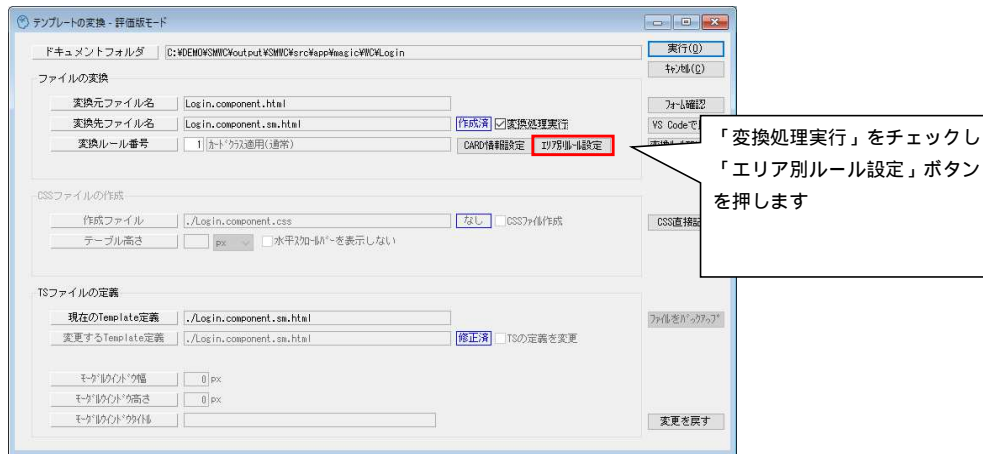


図2.4.3-8 エリア別ルール設定ボタン



図2.4.3-9 エリア別変換ルール設定

エリア別変換ルール設定

項目	説明
#	MAGICから出力した変換元のテンプレートを読み込み、分割したエリア番号を表示します。
行範囲	変換元テンプレートファイルの行範囲を行数で表示します。「...」ボタンを押すとソースの該当位置(開始行)をVisual Source Codeで開きます。
区分	「タブ」「テーブル」「サブフォーム」かそれ以外の場合は「通常」と表示します。
変換ルール	変換ルール番号を指定することが可能です。「...」ボタンで一覧から選択可能。
除外	親の画面で変換ルール番号を指定している場合、適用を除外することが可能です。

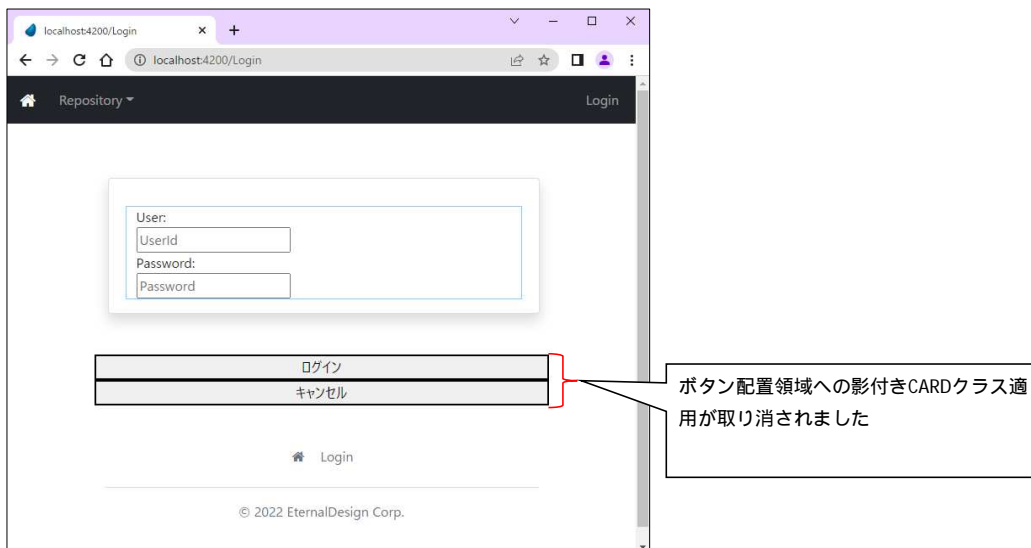


図2.4.3-10 エリア別の処理を適用したWeb画面

グリッドシステムを使ったフォーム配置

グリッドシステムを使うことにより、ラベルと入力エディットの位置を揃えることが可能です。
ウィンドウ幅が大きい場合はラベルとエディットをインラインで、小さい場合は縦に並べるような指定も可能です。

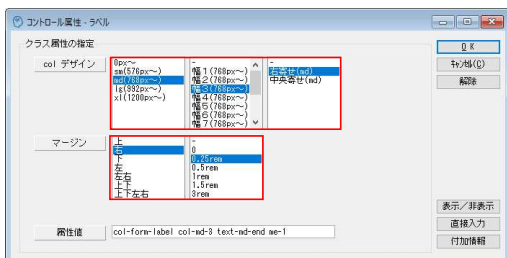
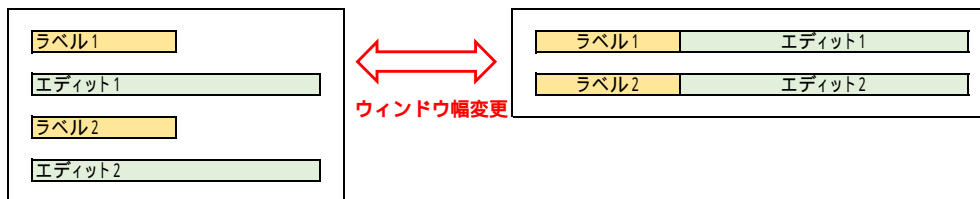


図2.4.3-11 ラベル1のコントロール属性入力例

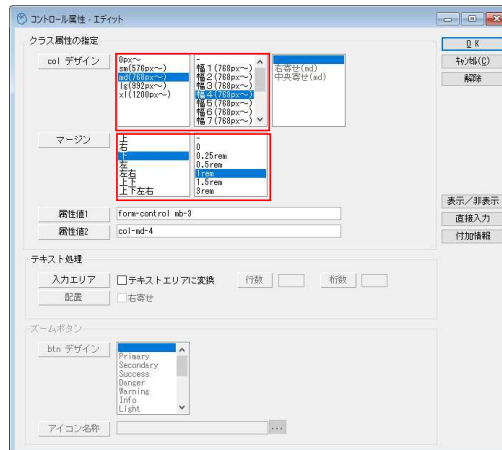


図2.4.3-12 エディット1のコントロール属性入力例

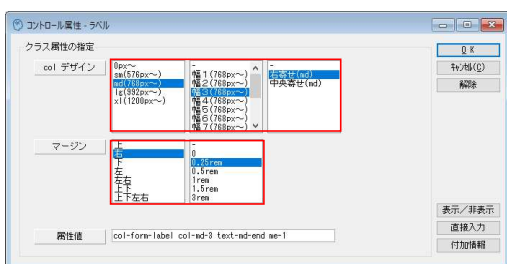


図2.4.3-13 ラベル2のコントロール属性入力例

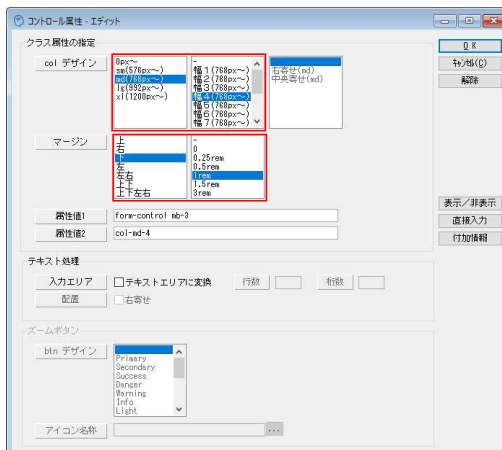


図2.4.3-14 エディット2のコントロール属性入力例

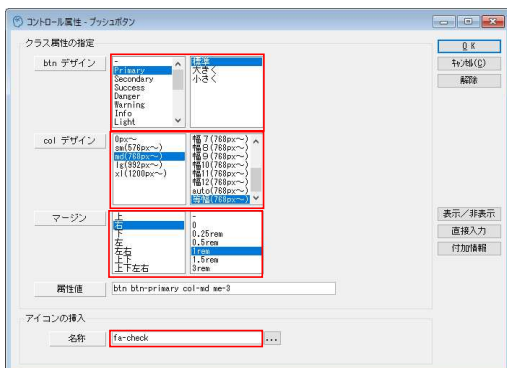


図2.4.3-15 プッシュボタン1のコントロール属性入力例

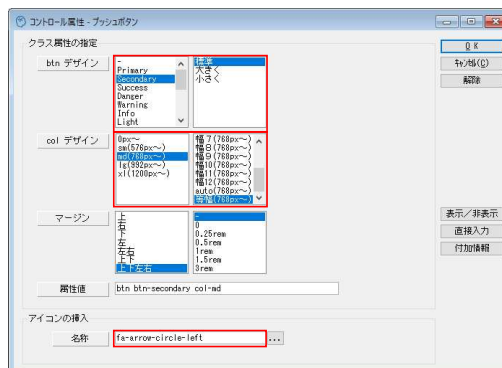


図2.4.3-16 プッシュボタン2のコントロール属性入力例

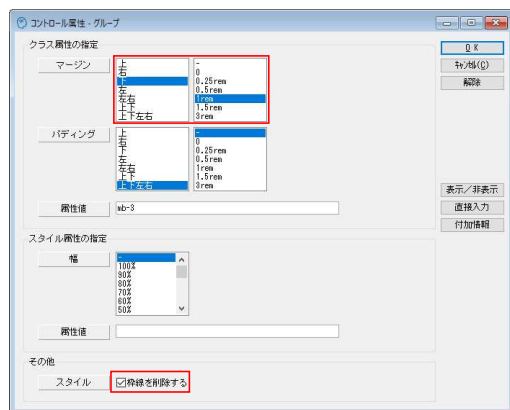


図2.4.3-17 グループのコントロール属性入力例

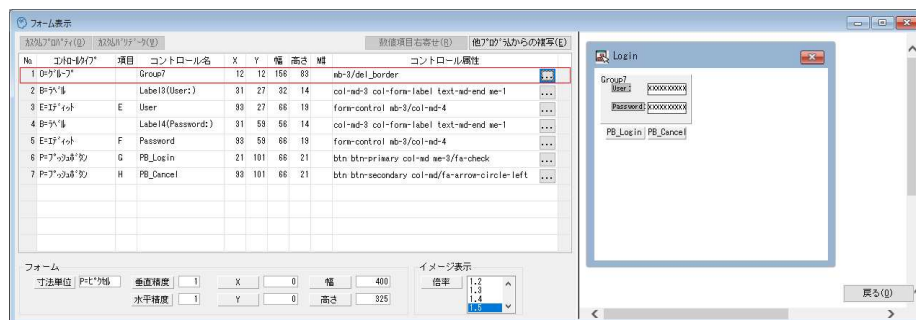


図2.4.3-18 コントロール属性の入力

動作を確認します。ウィンドウ幅によってダイナミックにレイアウトが変わります。

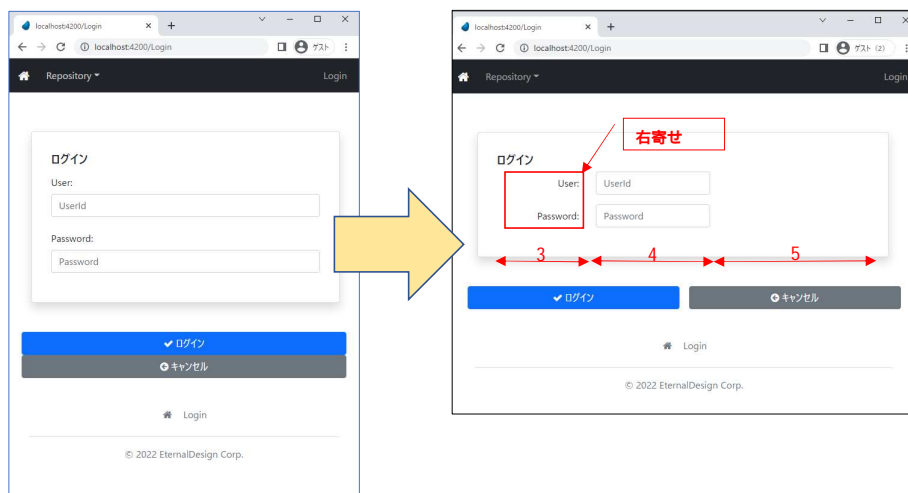


図2.4.3-19 グリッドシステムで配置したフォーム

2.4.4 オーバレイウィンドウの変換

フォームプロパティのウィンドウタイプを「O=オーバーレイ」に設定したプログラムは、プログラムコールから呼ばれた時に、モーダルダイアログとして表示します。MAGIC標準のオーバーレイを使用した時の幾つかの特性値をテンプレートの変換プログラムで指定することが可能です。

PI	プログラム名	ドキュメントフォルダ	フォルダ	タスク名	最終更新時刻	F#	フォーム名	OP	CV	ウィンドウタイプ	デフォルト/Web/タプル	処理
2	StartupProgram				2022/05/27 11:02	2	StartupProgram			D=デフォルト		<input checked="" type="checkbox"/> 変換
6	Home	<input checked="" type="checkbox"/>	WC		2022/05/27 11:02	2	Home			D=デフォルト		<input type="checkbox"/> 変換
7	Login	<input checked="" type="checkbox"/>	WC		2022/05/27 11:03	2	Login			D=デフォルト		<input checked="" type="checkbox"/> 変換
8	Logout	<input checked="" type="checkbox"/>	WC		2022/05/27 11:03	2	Logout			D=デフォルト		<input type="checkbox"/> 変換
9	Repository	<input checked="" type="checkbox"/>	WC		2022/05/27 11:03	2	Repository			D=デフォルト	R=繰り返し可能な項目	<input checked="" type="checkbox"/> 変換
10	LoginOverlay	<input type="checkbox"/>	WC		2022/05/27 11:03	2	LoginOverlay			O=オーバーレイ		<input checked="" type="checkbox"/> 変換
11	ProgramList	<input checked="" type="checkbox"/>	WC		2022/05/27 11:03	2	ProgramList			D=デフォルト		<input type="checkbox"/> 変換
11	ProgramTbl		WC	.1	2022/05/11 16:20	3	ProgramTbl			D=デフォルト	R=繰り返し可能な項目	<input type="checkbox"/> 変換
12	Program	<input checked="" type="checkbox"/>	WC		2022/06/24 20:12	2	Program			D=デフォルト		<input type="checkbox"/> 変換
12	ProgramTask		WC	.1	2022/05/27 11:03	3	ProgramTask			D=デフォルト		<input type="checkbox"/> 変換
12	DataView		WC	.1.1	2022/05/27 11:03	4	DataView			D=デフォルト	R=繰り返し可能な項目	<input type="checkbox"/> 変換
12	LogicView		WC	.1.2	2022/05/27 11:03	4	LogicView			D=デフォルト	R=繰り返し可能な項目	<input type="checkbox"/> 変換

オーバーレイタイプのプログラムをクリックし、「変換」ボタンを押します

図2.4.4-1 WebClientフォームでウィンドウタイプが「O=オーバーレイ」のフォームを選択

ドキュメントフォルダ: C:\DENON\SMC\output\SMC\src\app\Magic\WC\LoginOverlay

実行(O) / キャンセル(C)

ファイルの変換

変換元ファイル名: LoginOverlay.component.html

変換先ファイル名: LoginOverlay.component.sm.html 新規 ☐ 変換処理実行

変換ルール番号: 0 初期値に設定

VS Codeで比較 ☐ 変換後もTBL保存

フォーム確認

CSSファイルの作成

作成ファイル: ./LoginOverlay.component.css なし ☐ CSSファイル作成

テーブル高さ: px ☐ 水平方向に無限大を表示しない

CSS直接記述 ☐

TSファイルの定義

現在のTemplate定義: ./LoginOverlay.component.html

変更するTemplate定義: ./LoginOverlay.component.sm.html 相違 ☒ TSの定義を変更

モダリティウイダ幅: 300 px

モダリティウイダ高さ: 300 px

モダリティウイダタイトル: LoginOverlay

ファイル名を指定する

変更を戻す

オーバーレイのフォーム時は、オーバーレイ関連の項目が有効になります
入力した値でTSファイルを更新します

図2.4.4-2 「O=オーバーレイ」のフォーム選択時のテンプレートの変換

オーバーレイ関連項目		
項目	説明	明
モーダルウィンドウ幅	モーダルウィンドウの幅を指定します。	
モーダルウィンドウ高さ	モーダルウィンドウの高さを指定します。	
モーダルウィンドウタイトル	モーダルウィンドウのタイトルを指定します。	

TSファイルの定義

現在のTemplate定義: ./LoginOverlay.component.html

変更するTemplate定義: ./LoginOverlay.component.sm.html 相違 ☒ TSの定義を変更

モダリティウイダ幅: 600 px

モダリティウイダ高さ: 400 px

モダリティウイダタイトル: ログイン入力

図2.4.4-3 オーバレイ関連項目への入力例

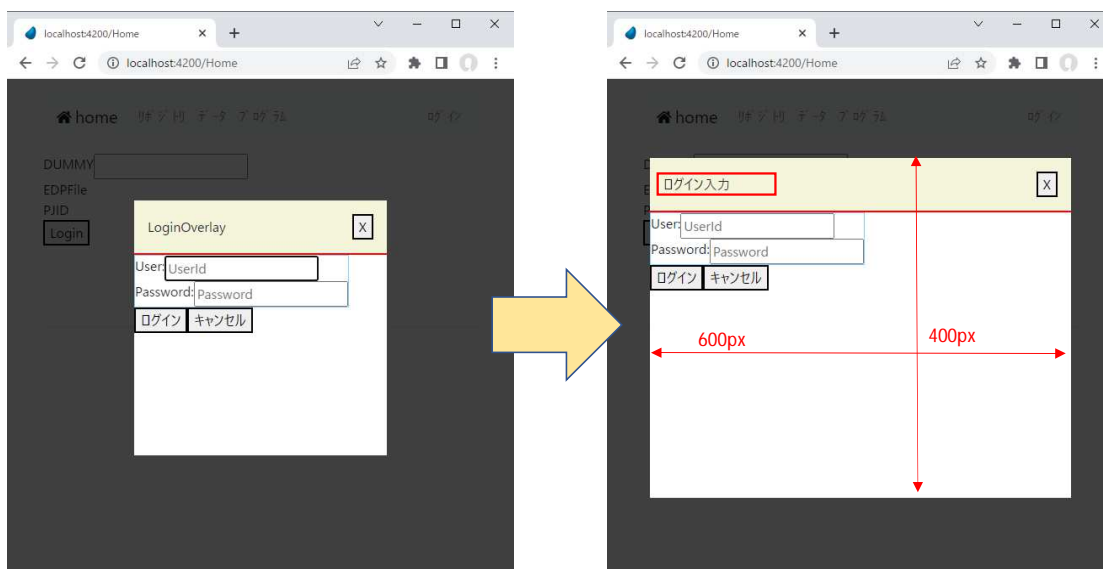


図2.4.4-4 オーバレイフォームのブラウザ表示

補足) 入力したコントロール属性のコピー

MAGIC xpaの開発版でプログラムをコピーして作成したフォームであれば、他のプログラムやフォームで入力済みのコントロール属性を複写することが可能です

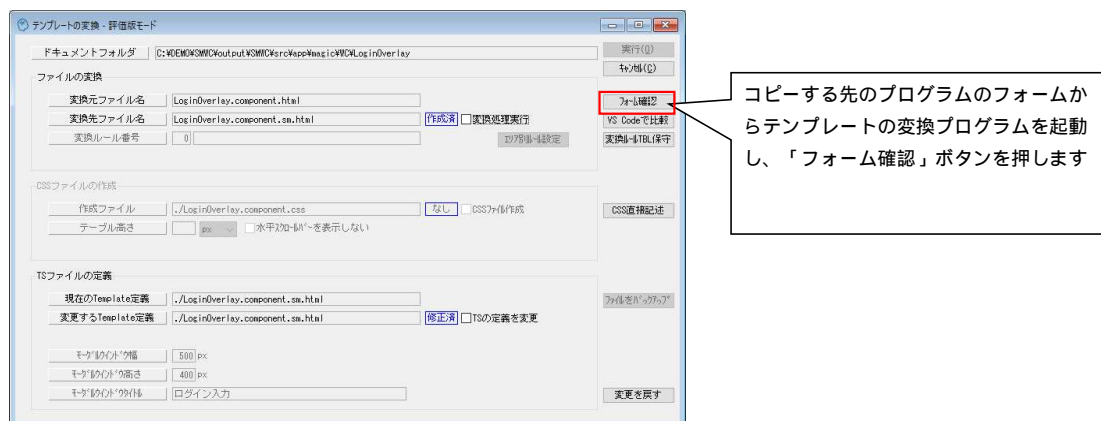


図2.4.4-5 コピーして作成したプログラムから入力済みの情報をコピー

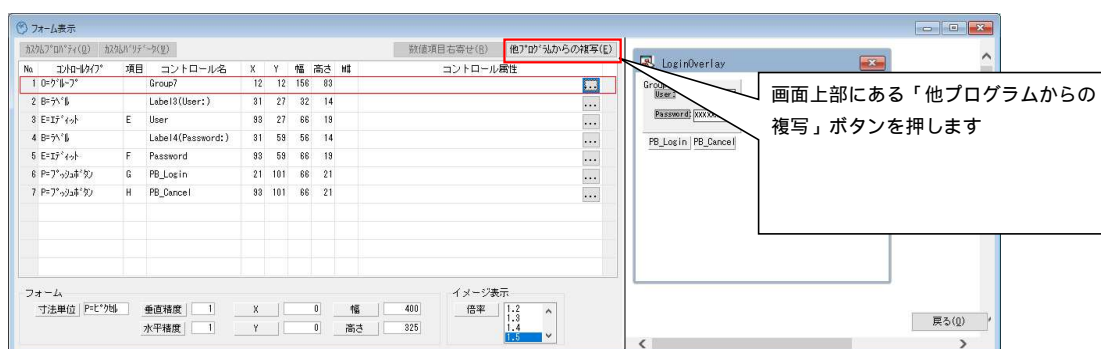


図2.4.4-6 「他プログラムからの複写」ボタンを使用して入力済みコントロール属性のコピー

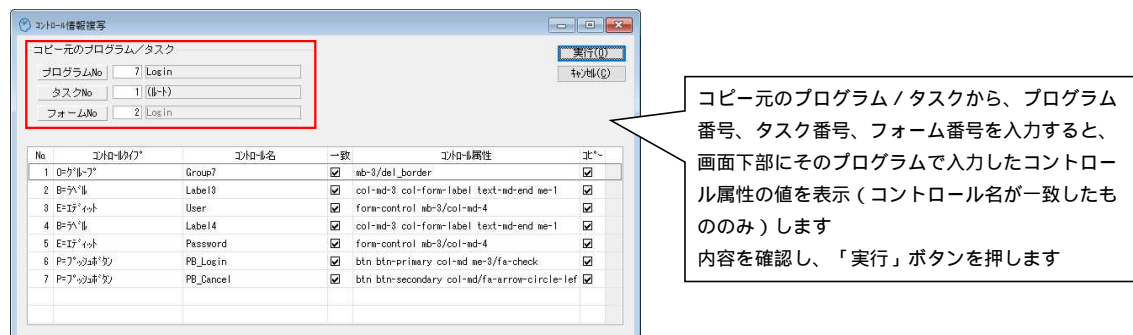


図2.4.4-6 コントロール情報複写プログラムで入力済み情報を取得

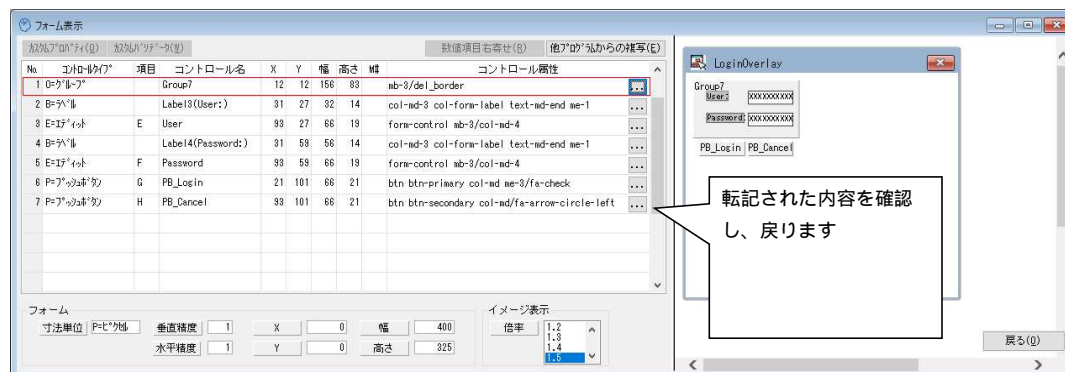


図2.4.4-7 複写された入力済み情報を確認

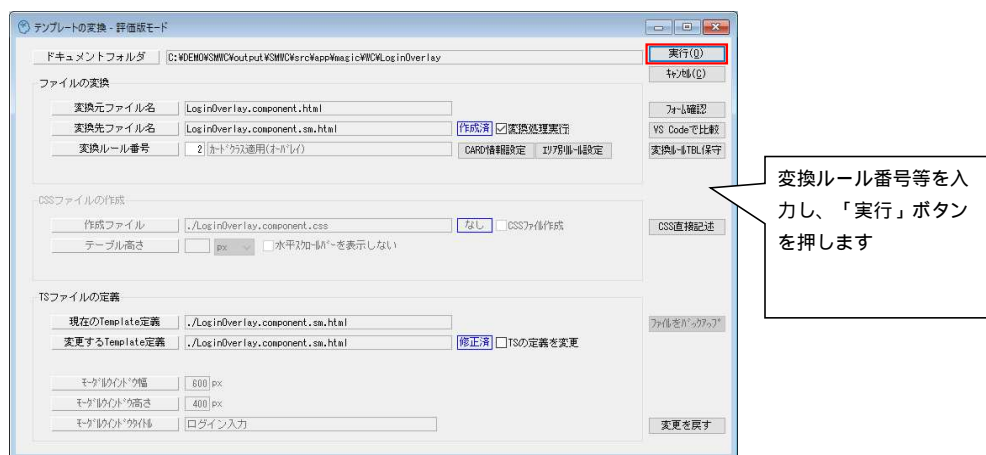


図2.4.4-8 テンプレートの変換(実行前)

「Login」ボタンを押すと、オーバーレイウィンドウに入力画面を表示します。

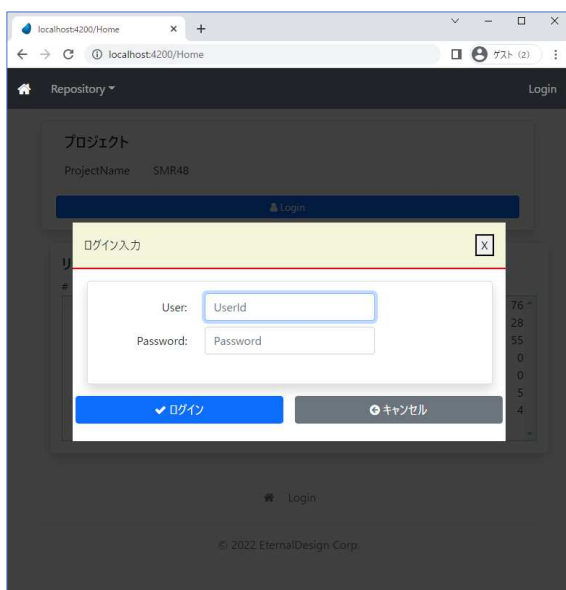


図2.4.4-9 ブラウザでの表示

2.4.5 カスタムプロパティの利用

カスタムプロパティは動的なユーザーインターフェスを伴うHTMLコンテンツを作成する際に有効な手段です。
「楽々ウェブクライアント」では、開発画面で設定したカスタムプロパティを簡単に視認し利用できる仕組みを用意しています。

MAGIC開発画面でのカスタムプロパティ定義

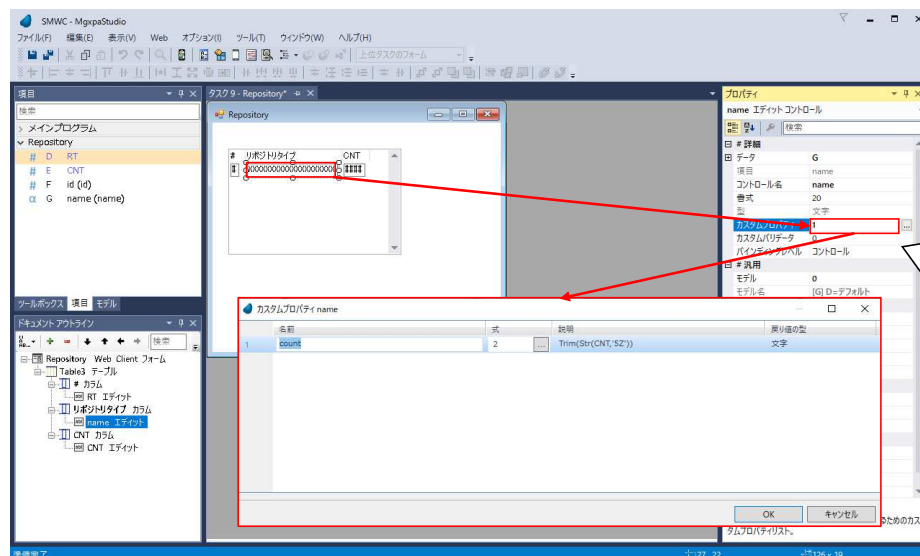


図2.4.5-1 MAGIC開発画面でのカスタムプロパティ

MAGIC開発画面では、
フォームやコントロール
に対してカスタムプロ
パティを設定します

しかし、フォームエ
ディタの深い場所にあ
るため、確認しづらい
かもしれません

楽々ウェブクライアントでのカスタムプロパティ有無確認

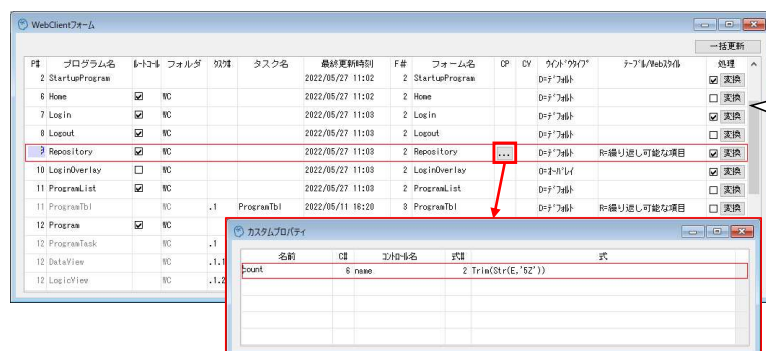


図2.4.5-2 楽々ウェブクライアントのWebClientフォーム一覧

楽々ウェブクライアントでは、ルー
ティング情報の「カスタムプロパ
ティ」ボタンや、WebClientフォーム
の「CP」列等で簡単に確認が可能で
す

HTMLコンテンツでの利用

件数をテーブルのCNT列で表示する代わりに、Bootstrapのバッジ(Badges)を使って表示してみましょう。
プログラム「Repository」の「フォーム確認」ボタンを押して、コントロールの一覧を表示します。
コントロール名「name」のエディット項目でコントロール属性を開きます。

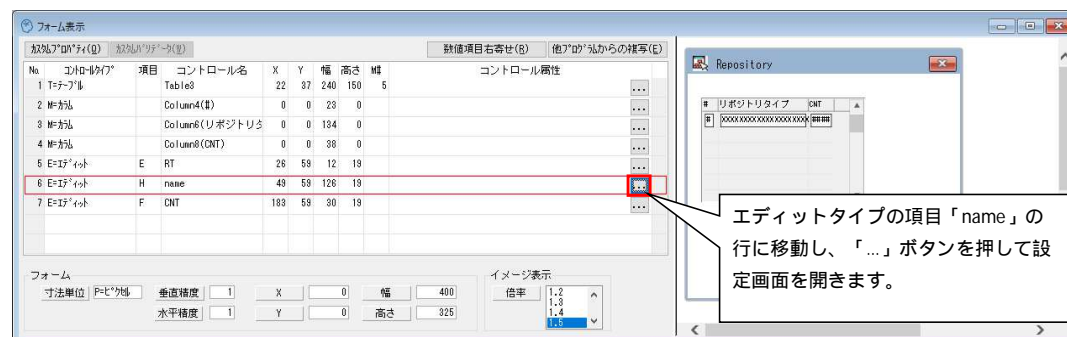


図2.4.5-3 フォーム表示

エディットタイプの項目「name」の
行に移動し、「...」ボタンを押して設
定画面を開きます。

「付加情報」ボタンを押し、入力ダイアログを表示します。

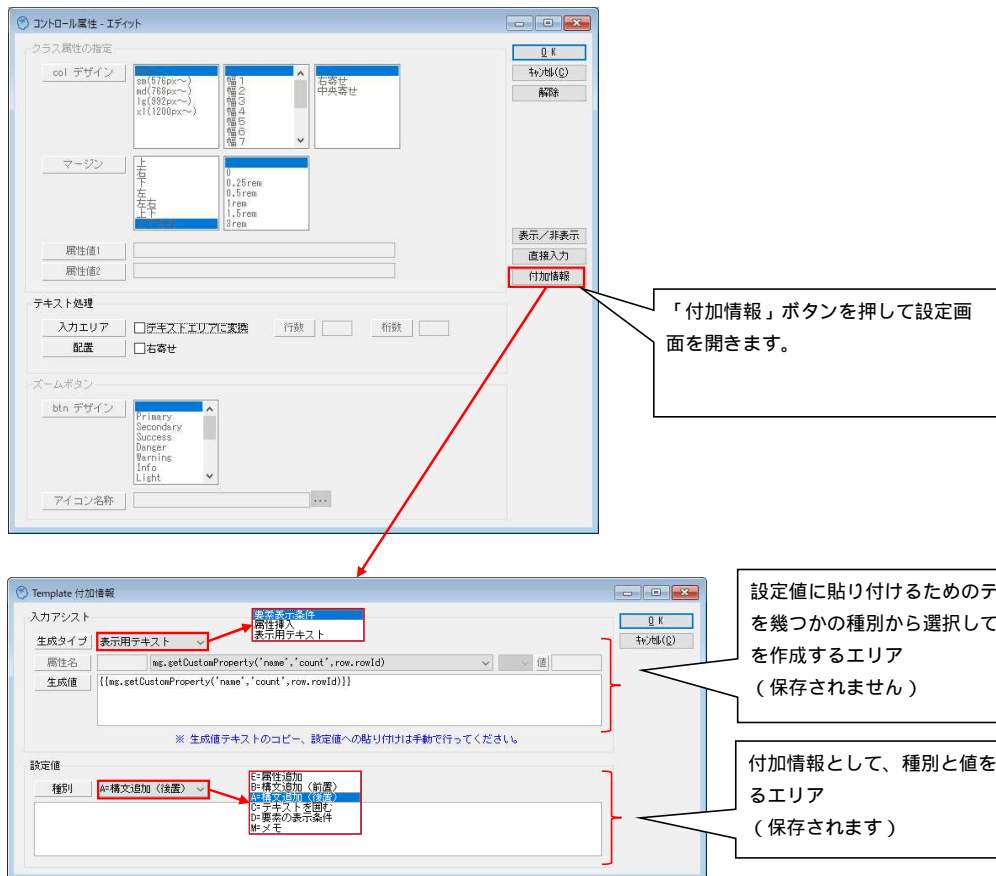


図2.4.5-4 Template付加情報

付加情報関連項目

項目	説	明
入力アシスト	設定値にコピーするための文字列を作るための支援を行います。	
生成タイプ	生成する文字列のタイプを「要素表示条件」「属性挿入」「表示用テキスト」から選択します。	
属性名	生成タイプが「属性挿入」の時、属性名を入力します。	
カスタムプロパティ値	フォーム内のカスタムプロパティの一覧がコンボボックスで表示されます。	
演算子	生成タイプが「要素表示条件」の時、演算子('','!','=','<','>','<=','>=')を選択します。	
値	生成タイプが「要素表示条件」の時、演算に使用する値を入力します。	
生成値	生成された文字列を表示します。	
設定値	種別と値を設定してHTMLに付加する文字情報を設定します。	
種別	設定した値の用途を決定します。	
	E=属性追加	属性を追加します。
	B=構文追加(前置)	構文を前に追加します。
	A=構文追加(後置)	構文を後に追加します。
	C=テキストを囲む	テキストを囲みます。テキスト部分は '%' で指定します。
	D=要素の表示条件	書式:<要素>%</要素>要素の表示条件を指定します。
値	M=メモ	処理しません。
		HTMLに挿入される文字列を設定します。入力アシストで作成した生成値のコピーは手動で行います。

入力例

- 1) 「種別」は「A=構文追加(後置)」を選択します。
- 2) テキストエリアに「 」を入力します。
- 3) class属性「"badge bg-success ms-3"」を の位置に追加します。
- 4) 入力アシスト機能を使い、「要素表示条件」で「*ngIf="mg.getCustomProperty('name','count',row.rowId)!=0"」を作成し、位置 に挿入します。
- 5) 入力アシスト機能を追加、「表示用テキスト」で「{{mg.getCustomProperty('name','count',row.rowId)}}」を作成し、位置 に挿入します。

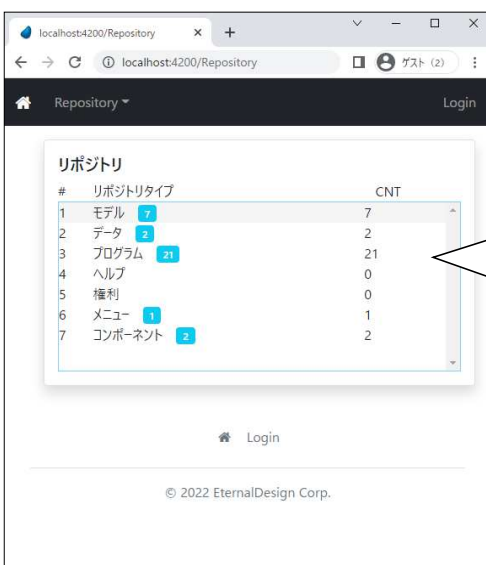
設定例

```
<span
  *ngIf="mg.getCustomProperty('name','count',row.rowId)!=0" class="badge bg-success ms-3">
  {{mg.getCustomProperty('name','count',row.rowId)}}
</span>
```

Bootstrap 4.xの時は「class="badge badge-info ml-3"」のように指定します。

更にこの部分を別途定義したカスタムプロパティを使って定義し、色 (badge-xxxx) を変える等の処理も可能です。
(入力アシストの「属性挿入」を使用)

図2.4.5-5 入力アシストを使用したTemplate付加情報の設定



件数が1以上の時にバッジが表示されていることが分かります。

「CNT」列は不要ですが、数値を確認するために表示しています。
列の表示有無や幅の指定も「col デザイン」の指定で行うことが可能です。

図2.4.5-6 バッジ表示を実現したWeb画面の例



ソースに挿入されたカスタムプロパティによる構文 (labelタグの後ろに挿入 : 「構文追加(後置)」)

図2.4.5-7 Visual Studio Codeによるソースの確認

2.4.6 ズームボタン

ズームボタンはエディットコントロールの「ズームボタンを表示」プロパティで設定するMAGICの機能です。下図は、モデルが指定されている数値項目をフォームに配置した時のフォームエディタ上の画面です。

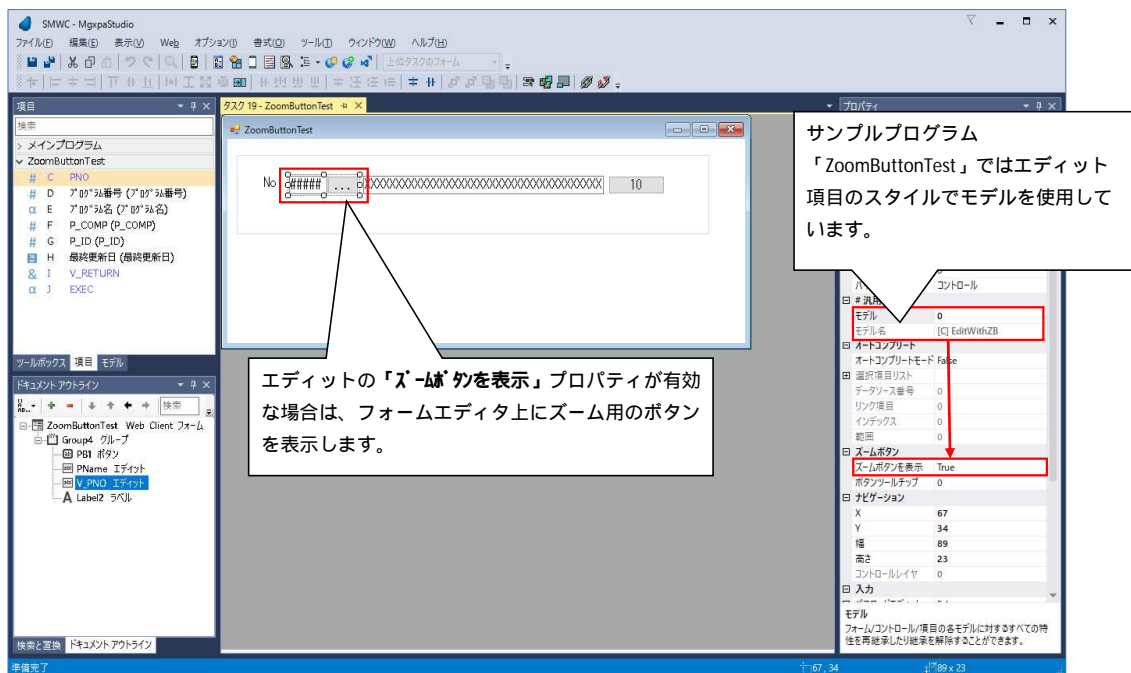


図2.4.6-1 MAGICフォームエディタ画面 (ズームボタンを表示したエディットコントロールを配置)

< 補足 > ズームボタンの表示方法

ズームボタンを表示させる方法には幾つかあります。優先順位的には下記の通りになります。

- ・ フォームエディタ上でコントロールのプロパティを直接指定
- ・ フォームエディタ上でモデル (ズームボタンを表示) を指定
- ・ データ項目の該当のスタイルでプロパティを直接指定
- ・ データ項目の該当のスタイルでモデル (ズームボタンを表示) を指定

楽々ウェブクライアントでの変換

まずはそのまま素の状態 (変換ルールの指定なし、コントロール属性の設定なし) で変換してみます。

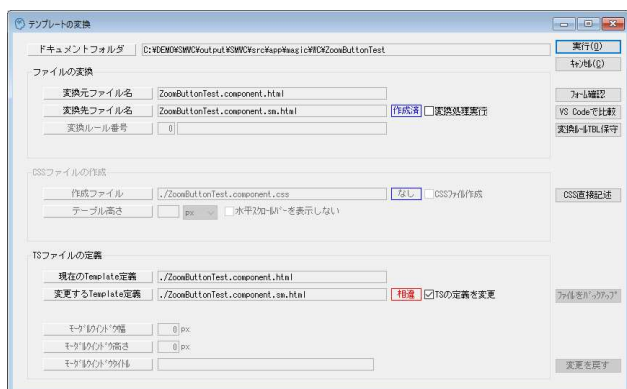


図2.4.6-2 テンプレートの変換 (変換ルールを指定せずに変換を行う)

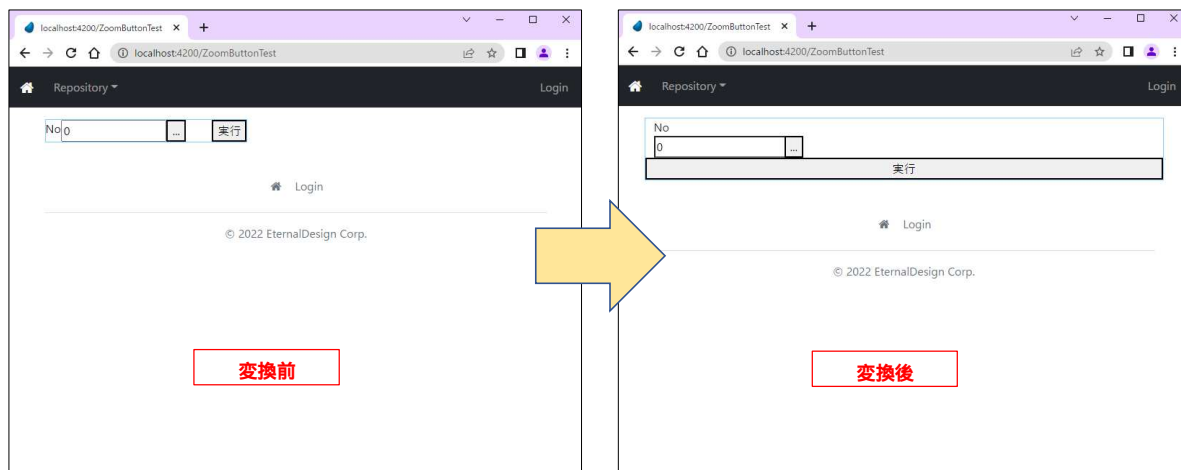


図2.4.6-3 変換結果

見栄えはあまり大きく変わりませんが、ソースを確認するとズームボタン付きのエディットコントロールの部分は下図のように変わっていることが分かります。

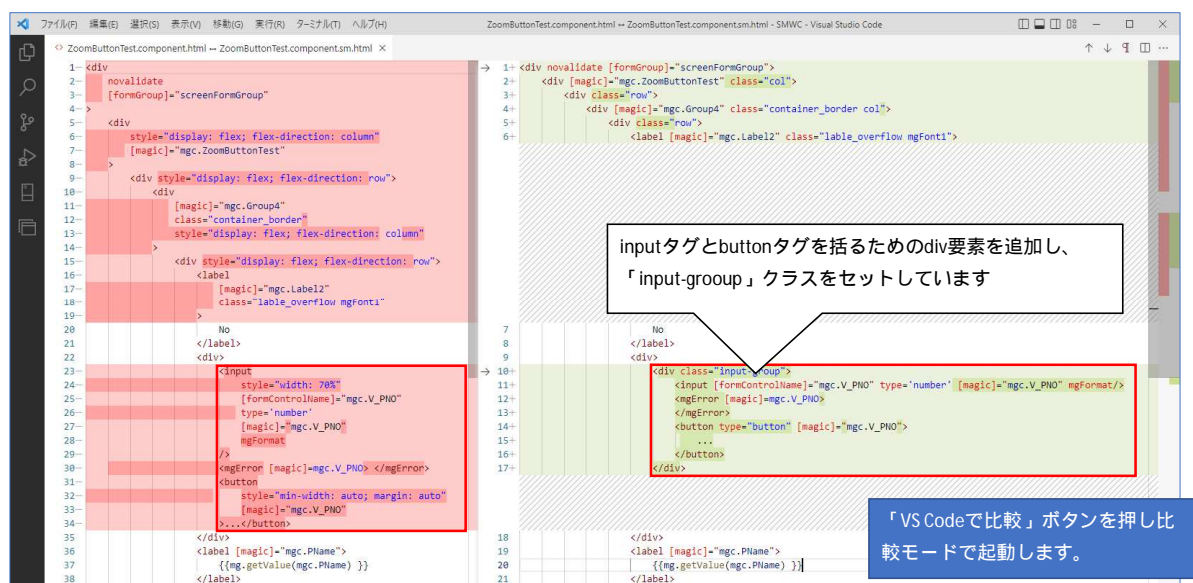


図2.4.6-4 VS Codeによる比較結果

コントロール属性画面

ボタンのデザインや、アイコンをコントロール属性の設定ダイアログで指定することが可能です。
ズームボタンが有効な場合は、赤枠内のコントロールが有効になります。

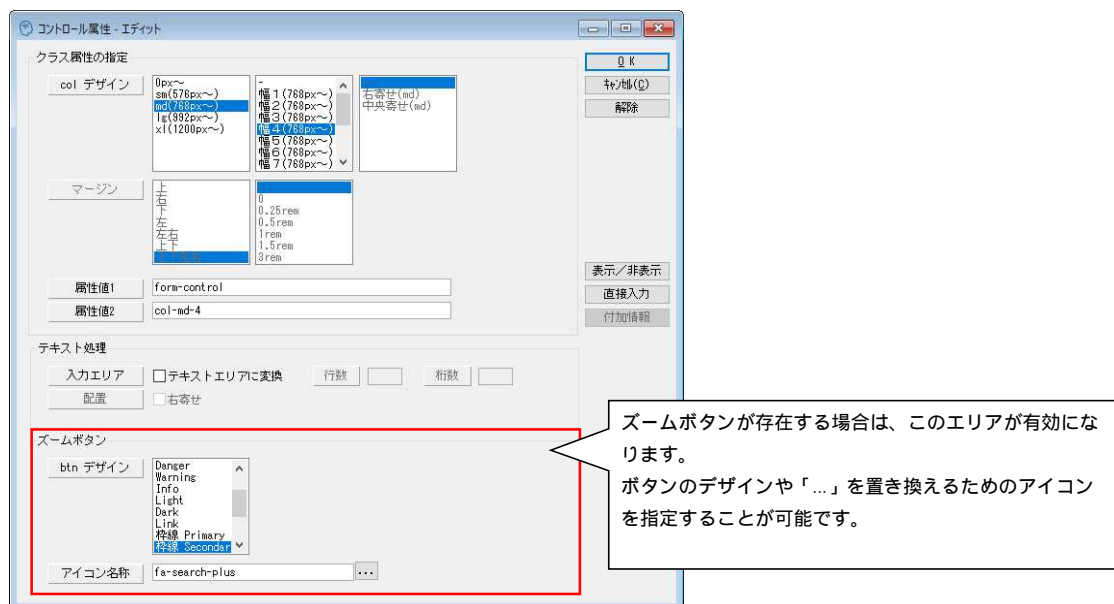


図2.4.6-5 VS Codeによる比較結果

デザイン例

ボタンデザインを変更した時のデザイン例を示します。

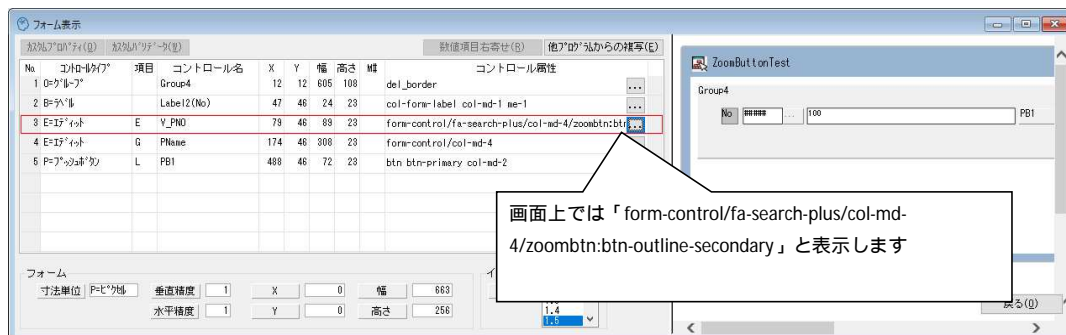


図2.4.6-6 「フォーム表示」での属性値表示

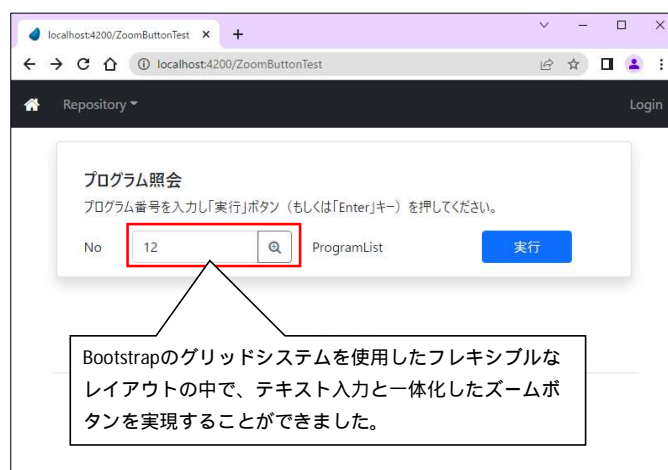


図2.4.6-7 ズームボタンの変換実行結果

2.4.7 グリッドシステムの応用

Bootstrapのグリッドシステムの利用方法として、 テーブルのカラムに適用する方法(2.4.2「テーブルの変換」を参照)と、 グループに関連付けたコントロールに適用する方法(2.4.3「テーブル定義のないタスクフォームの変換」を参照)について説明してきました。
 楽々ウェブクライアントVer1.02では、更に上位のレベルでテーブルとグループコントロール(もしくは、サブフォーム等)間の配置にグリッドシステムを利用することができるようになっています。

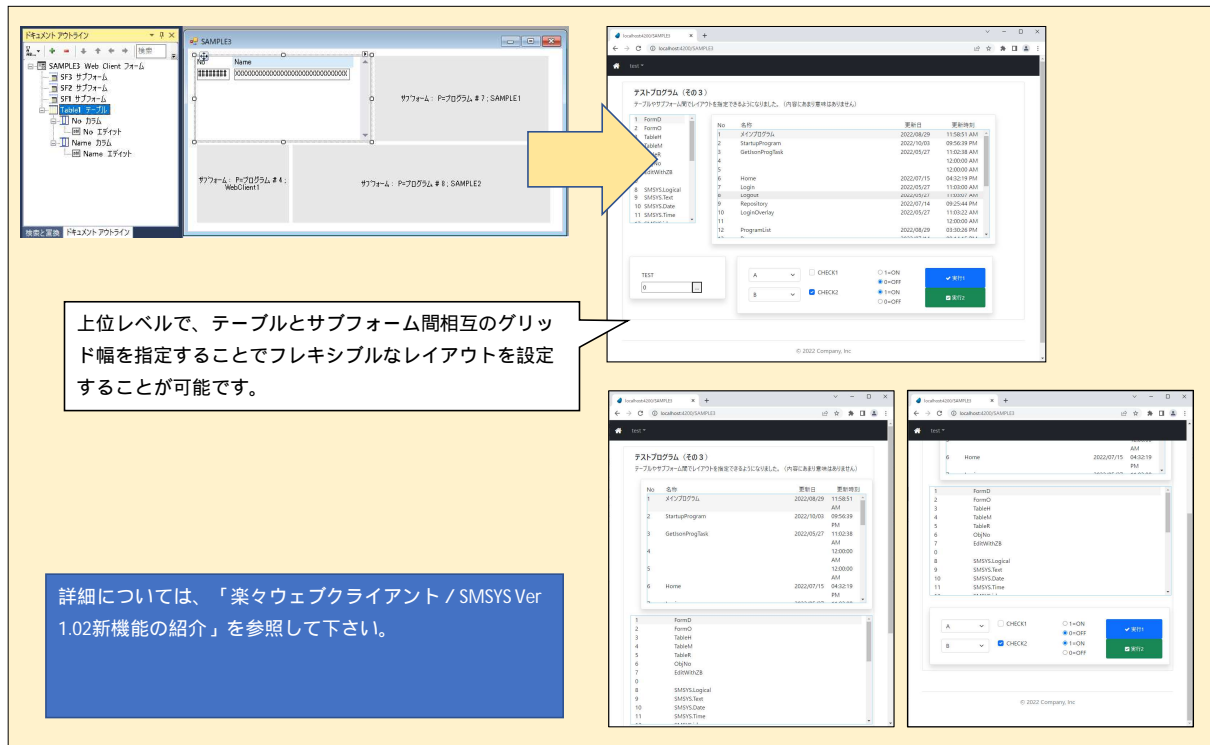


図2.4.7-1 テーブルとサブフォームにグリッドシステムを適用した例

2.4.8 その他の機能

Visual Studio Code連携

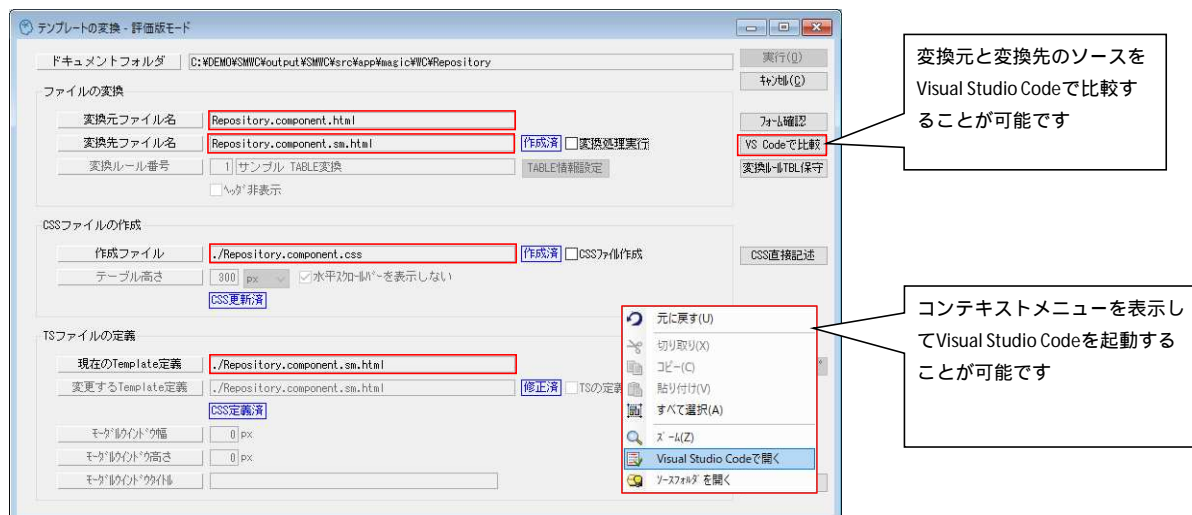


図2.4.8-1 Visual Studio Code連携

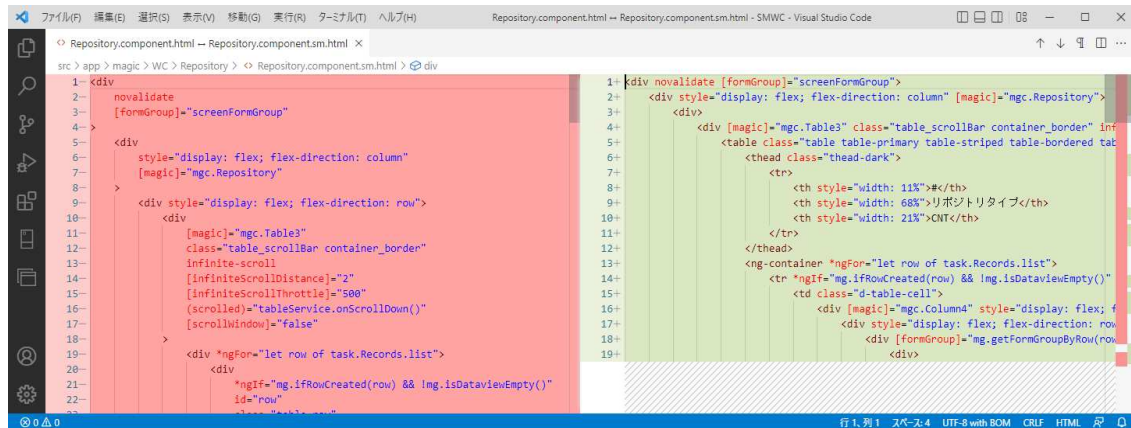
イ) コンテキストメニュー

コンテキストメニューのVisual Studio CodeでAngularソースを開きます。
メニューを呼び出したコントロールによって、呼び出すファイルが変わります。

変換元ファイル名	変換元のファイル (MAGICが出力したファイル)
変換先ファイル名	(出力済み) のとき 変換先のファイル
CSSファイル	(出力済み) のとき CSSファイル
「TSファイルの定義」	TSファイル
現在のTemplate定義	
その他の箇所	TSファイルで指定されたTemplateファイル

ロ) 「VS Codeで比較」ボタン

変換済みであればボタンが有効になります。オリジナルのソースと比較して結果を確認することが可能です。



TSファイルのバックアップ

Visual Studio Code等でTSファイルを直接的に編集したときのバックアップファイルを保存します。

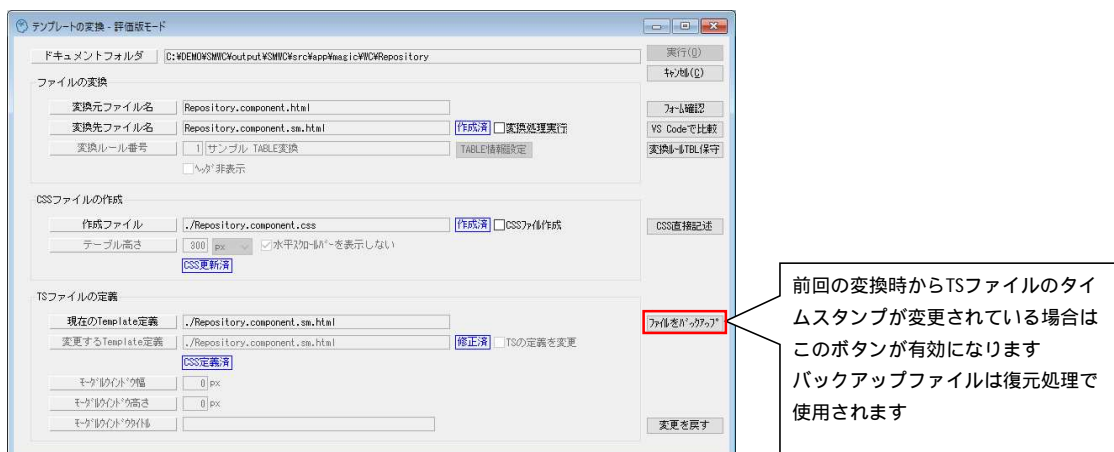


図2.4.8-3 TSファイルのバックアップ

変更を戻す

WebClient関連の編集データはメニュー画面の「初期化」ボタンを押してもリセットしない仕様になっています。何らかの理由でデータを削除する必要がある場合は、フォーム単位で個別に削除する必要があります。「変更を戻す」ボタンでデータの削除等を行います。

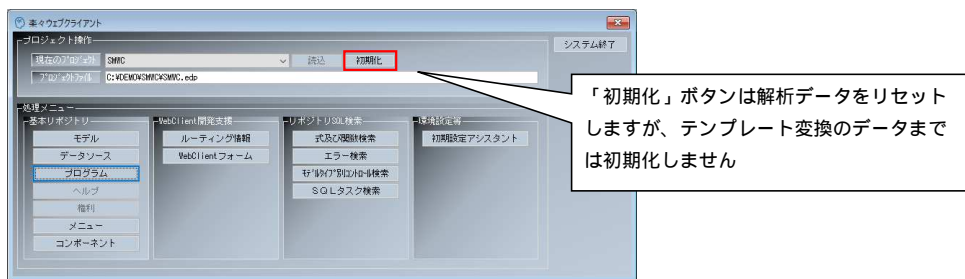


図2.4.8-4 初期化ボタン

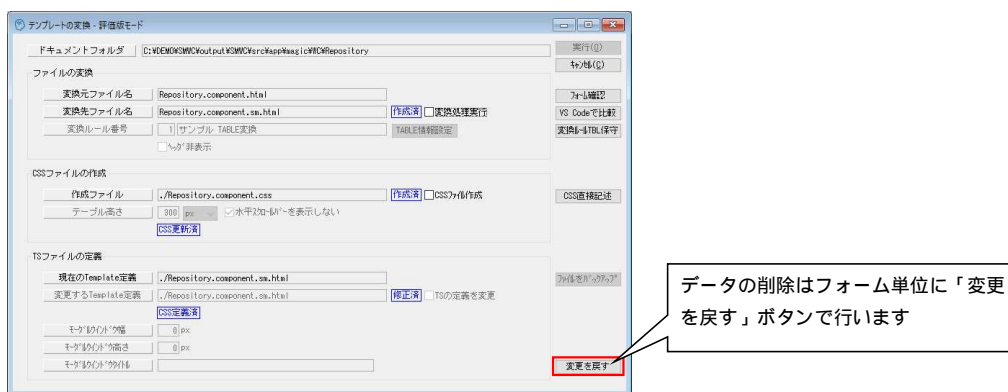


図2.4.8-5 「変更を戻す」ボタン



図2.4.8-6 「変更を戻す」ダイアログ

「変更を戻す」画面の項目

項目	説明
Templateを戻す	チェック時、MAGICが出力した(変換元の)テンプレートにTSファイルの内容を書き変えます。
CSSを除外する	チェック時、TSファイルのスタイルシートの定義を除外します。(作成済のCSSがあるとき)
フォーム編集情報を削除	チェック時、変換ルール番号、エリア別ルール設定、CARD / TABLEのクラス情報等の指定した内容を削除します。
コントロール属性を削除	チェック時、入力済みのコントロール情報を削除します。

変換ルールTBL保守

変換ルールTBLボタンを押すと変換ルールの内容を編集が可能です。

正規版で有効な機能です。

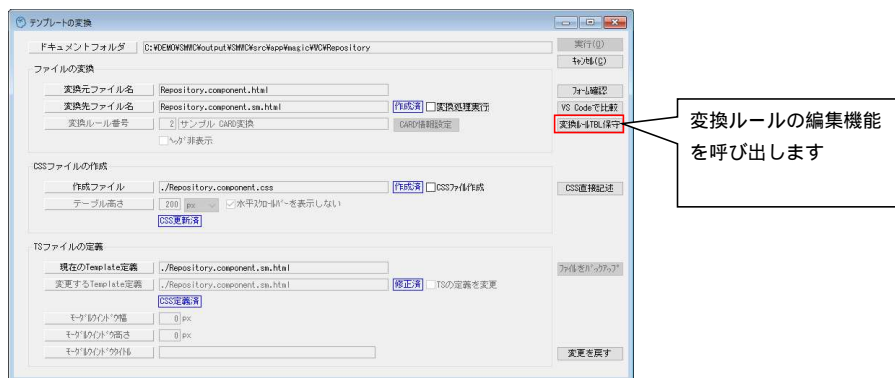
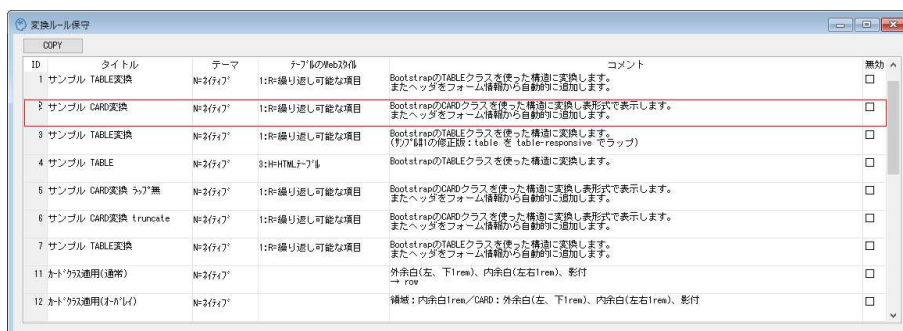


図2.4.8-7 「変換ルールTBL保守」ボタン



親の画面で、「変換ルール番号」が入力されている時は、呼出し時、その変換ルールに位置付けします。

変換ルール保守の項目

項目	説明
「COPY」ボタン	既存のルールをコピーして新しいルールを押すときはこのボタンを押して下さい。'図2.4.6-9'、
ID	変換ルールを決める一意の番号を入力します。
タイトル	表示するタイトルを入力します。
テーマ	通常は「N=ネイティブ」を入力します。利用するWebアプリケーションのテーマに合わせます。
テーブルのスタイル	テーブル用の変換ルールを設定する場合は、「R=繰り返し可能な項目」「M=マテリアルデザイングリッド」「H=HTMLテーブル」から選択します。
コメント	表示するコメントを入力します。
無効	無効にチェックすると選択画面から非表示になります。
「F5」キー	明細編集画面を開きます。'図2.4.6-10'、

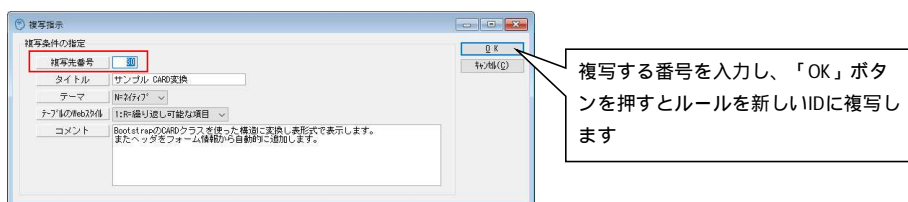


図2.4.8-9 複写指示

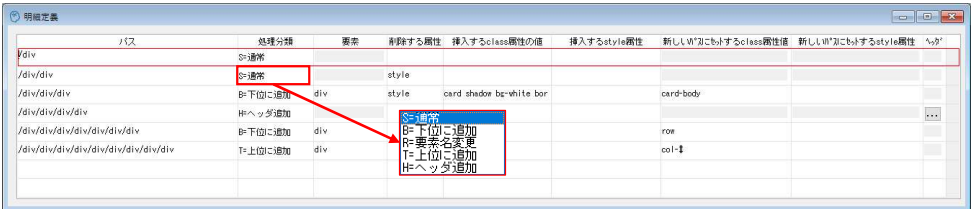


図2.4.8-10 明細定義

明細定義の項目		
項目	説	明
パス	HTMLのパスを指定します。 上位の階層の処理分類で追加 / 変更を指定した時は変更後のパスを指定します。	
処理分類	パスに対する処理を選択します。 S=通常 パスの処理が不要な場合に指定します。 B=下位に追加 下位にパスを追加する場合に指定します。 R=要素名変更 現在の要素名を変更する場合に指定します。 T=上位に追加 上位にパスを追加する場合に指定します。 H=ヘッダ追加 テーブルのヘッダを追加する場合に指定します。	
要素	処理分類で追加 / 変更を指定した場合に要素名を入力します。	
削除する属性	属性を削除したい場合に名称を入力します。	
挿入するclass属性の値	class属性を指定したい場合に値を入力します。	
挿入するstyle属性の値	style属性を指定したい場合に値を入力します。	
新しいパスにセットするclass属性の値	新しいパスにclass属性を指定したい場合に値を入力します。処理分類で「下位に追加」「上位に追加」を選択したときに有効になります。	
新しいパスにセットするstyle属性の値	新しいパスにstyle属性を指定したい場合に値を入力します。処理分類で「下位に追加」「上位に追加」を選択したときに有効になります。	
ヘッダ	ヘッダの定義画面を開きます。(処理分類がヘッダ追加のときのみ)「 図2.4.6-11 」	



図2.4.8-11 TBLヘッダ

class属性の「col-#」はグリッドシステムのグリッド幅として処理します。
tableクラスの場合はstyle属性の「width: %」で指定します。

TBLヘッダの項目		
項目	説	明
パス	HTMLのパスを指定します。	
属性	設定する属性を入力します。	
値	属性に対する値を入力します。	
項目でループ	繰り返し項目にチェックします。	

CSS直接記述

「CSS直接記述」ボタンを押すことによりスタイルシートの自由な入力編集が可能になります。

正規版で有効な機能です。

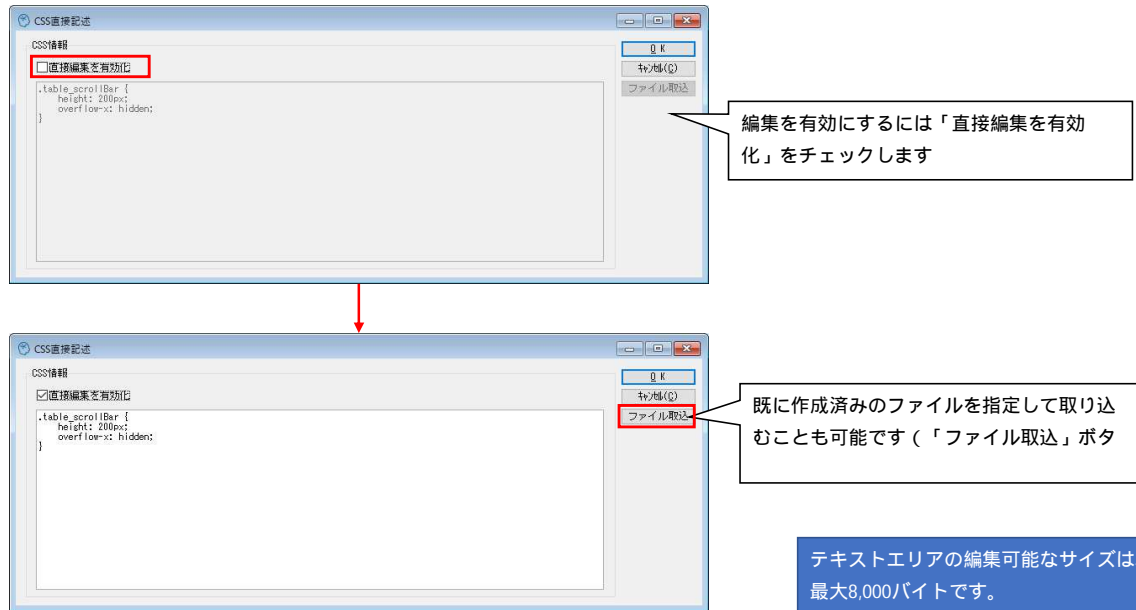


図2.4.8-12 CSS直接記述

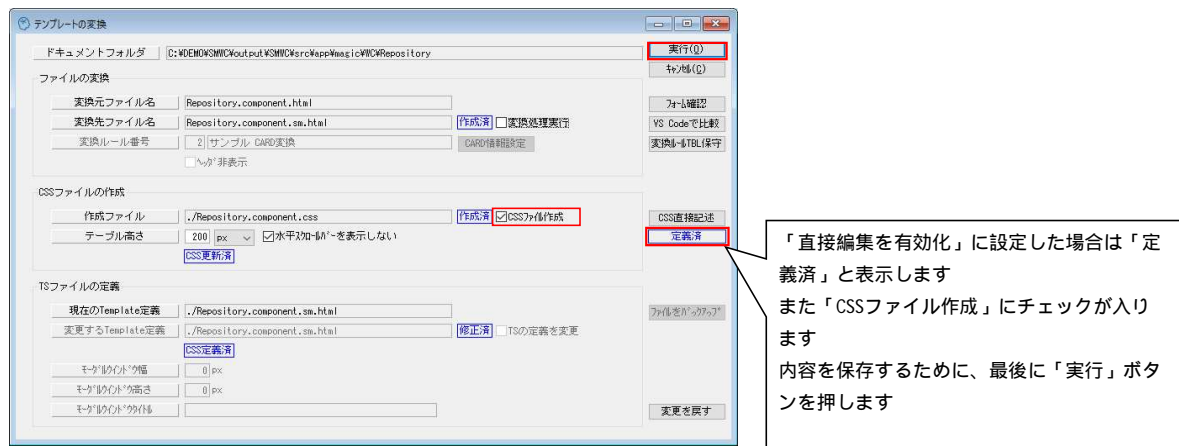


図2.4.8-13 CSS直接記述有効時のテンプレートの変換ダイアログ

2.5 サンプルプロジェクト

2.5.1 SMSYSとサンプルプロジェクトSMWC

サンプルプロジェクトSMWSはコンポーネントとしてのSMSYSを利用することにより、MAGICのソースにアクセスしてリポジトリ内容を表示するものです。コンポーネントSMSYSはプロジェクト解析処理関連のプログラム、解析結果データを提供しています。

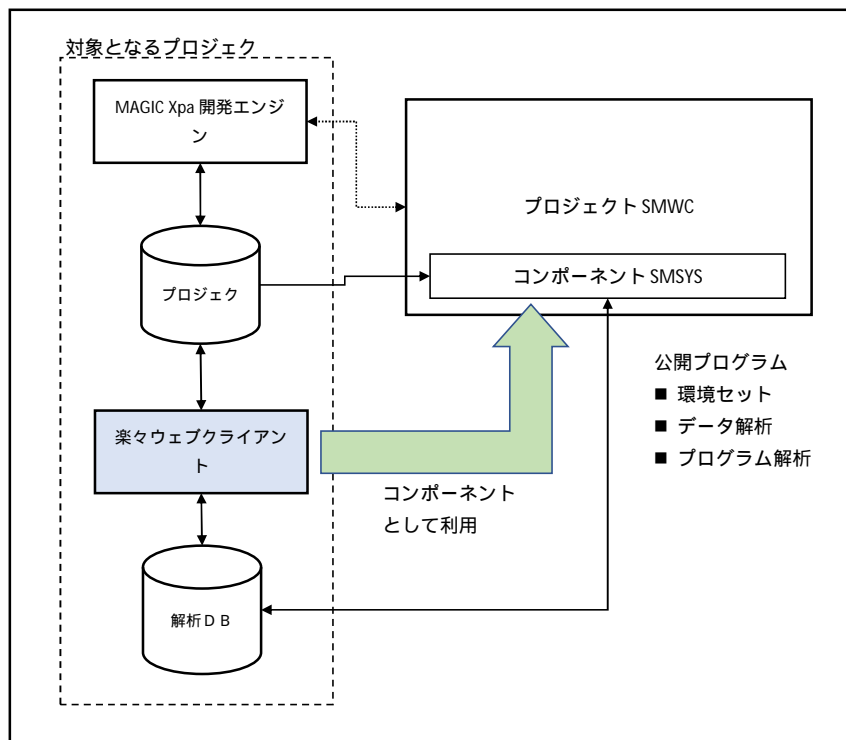


図2.5.1-1 サンプルプロジェクトSMWCはコンポーネントにSMSYSを利用

2.5.2 サンプルプロジェクトSMWCのプロジェクト切り替え方法

サンプルプロジェクトSMWSはデータベースの設定から対象となるプロジェクトを判定します。対象となるプロジェクトの指定は下記のいずれかの手順で行ってください。

- ・ 実行版で切り替える
- ・ データベースの設定を変更する

実行版で切り替える

デスクトップに作成したショートカットで楽々ウェブクライアントを起動
「現在のプロジェクト」コンボボックスから対象となるプロジェクトを選択し表示されるダイアログに従って再起動
開発版を再起動してプロジェクトSMWSを実行

データベースの設定を変更する

開発版を起動、もしくは開いているプロジェクトを閉じ、データベースの設定を開く
「SMSYSDB」のDB名、データベースサーバの値を書き変える
プロジェクトSMWSを開き、プロジェクト実行

サンプルプロジェクトSMWCのデザインを並行的に編集する場合は、ユーザ定義開発メニューから起動してください。

2.5.3 サンプルプロジェクト、デザイン例

サンプルプロジェクトのデザイン例を示します。
初期値から変更した箇所のみ記載しています。
コントロール属性については、コントロール一覧画面に表示された結果のみ表示しています。(若干分かり辛いかもしれませんが)

1) Home / Repository

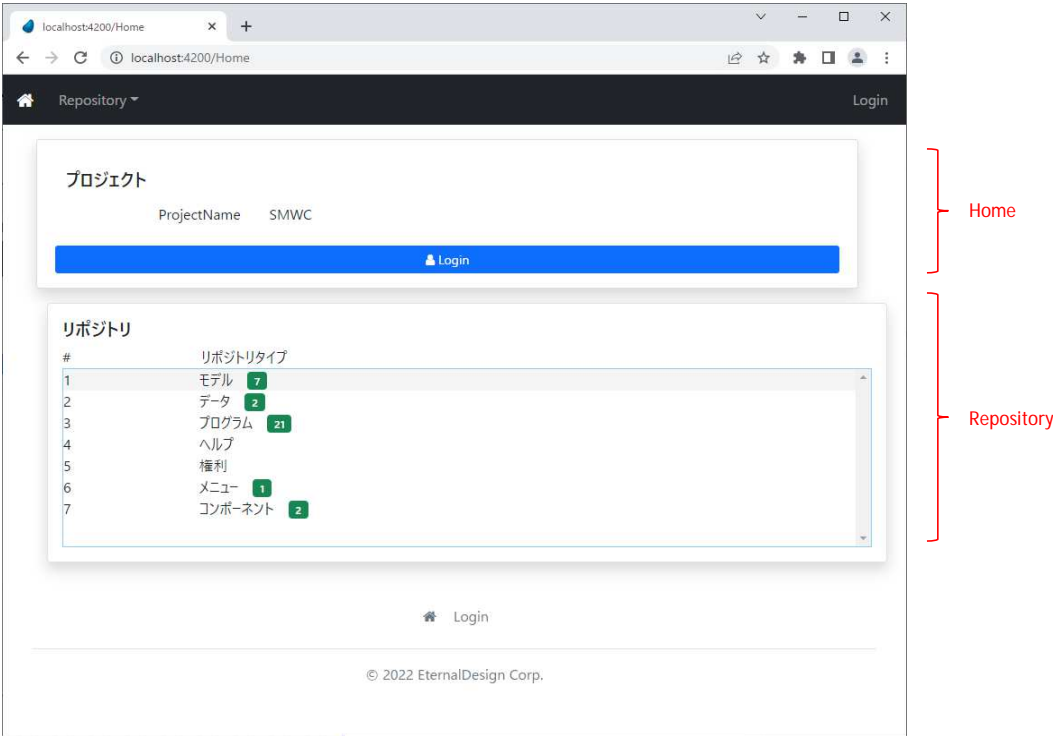


図2.5.3-1home画面のデザイン例

Home

変換ルール番号 1:カードクラス適用(通常)
CARD情報設定 タイトル:プロジェクト
エリア別ルール設定 エリア2を除外

コントロール属性

コントロールタイプ	項目	コントロール名	x	y	幅	高さ	設定値
1 O=グループ		Group1	12	12	301	108	del border
2 B=ラベル		Label1	21	30	74	14	col-md-3 col-form-label text-md-end me-2
3 E=エディット	E	ProjectName	101	30	186	19	form-control mb-3/col-md-3
4 P=パッシュボタン	F	PB_LOGIN	23	71	72	21	btn btn-primary btn-sm/fa-user/(付加情報) 付加情報で'要素の表示条件'を選択し、下記を設定 *ngIf="mg.getCustomProperty('Home','user')==0"
5 U=サブフォーム		SF1	12	126	192	192	width: 100%;

Repository

変換ルール番号 2:サンプル CARD変換
CARD情報設定 タイトル:リポジトリ
テーブル高さ 200px
水平スクロールバー 表示しない

コントロール属性

コントロールタイプ	項目	コントロール名	x	y	幅	高さ	設定値
1 T=テーブル		Table3	22	37	240	150	
2 M=カラム		Column4	0	0	23	0	d-block col-2
3 M=カラム		Column6	0	0	134	0	d-block col-5
4 M=カラム		Column8	0	0	38	0	d-none
5 E=エディット	E	RT	26	59	12	19	(付加情報)
6 E=エディット	H	name	49	59	126	19	付加情報で'構文追加(後置)'を選択し、下記を設定 {mg.getCustomProperty('name','count',row.rowId)}}
7 E=エディット	F	CNT	183	59	30	19	

2) Login

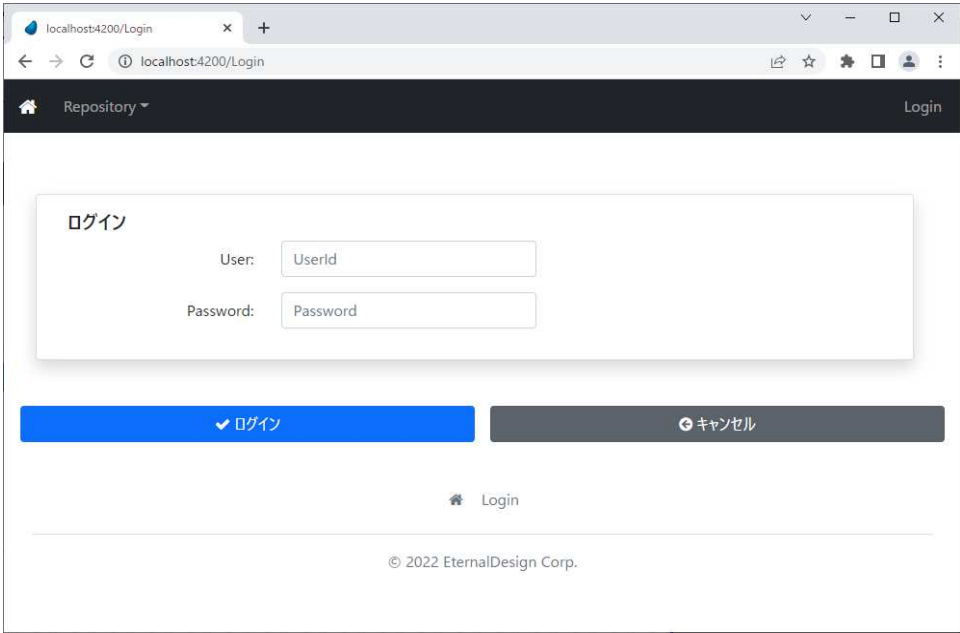


図2.5.3-2 ログイン画面のデザイン例

Login

変換ルール番号 1:カードクラス適用(通常)
CARD情報設定 タイトル:ログイン
マージン:上3remに変更
マージン:下3remに変更

コントロール属性							
コントロールタイプ	項目	コントロール名	x	y	幅	高さ	設定値
1 O=グループ		Group7	12	12	156	83	mb-3/del border
2 B=ラベル		Label3	31	27	32	14	col-md-3 col-form-label text-md-end me-1
3 E=テキスト	E	User	93	27	66	19	form-control mb-3/col-md-4
4 B=ラベル		Label4	31	59	56	14	col-md-3 col-form-label text-md-end me-1
5 E=テキスト	F	Password	93	59	66	19	form-control/col-md-4
6 P=プッシュボタン	G	PB Login	21	101	66	21	btn btn-primary col-md me-3/fa-check
7 P=プッシュボタン	H	PB Cancel	93	101	66	21	btn btn-secondary col-md/fa-arrow-circle-left

3) ProgramList / ProgramTbl

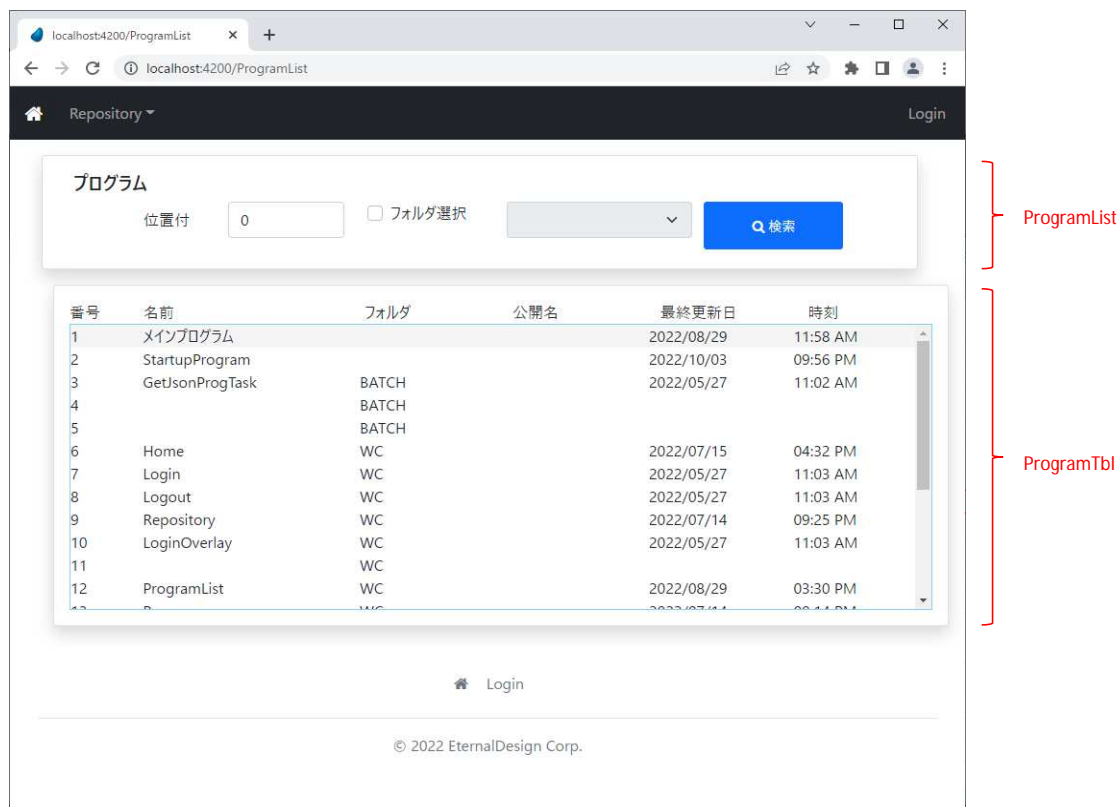


図2.5.3-3 プログラム一覧画面のデザイン例

ProgramList

変換ルール番号 1: カードクラス適用(通常)
 CARD情報設定 タイトル: プログラム
 エリア別ルール設定 エリア2を除外

コントロール属性

コントロールタイプ	項目	コントロール名	x	y	幅	高さ	設定値
1 O=グループ		Group7	12	12	470	47	del border
2 B=ラベル		Label2	26	29	50	14	col-md-2 col-form-label text-md-end me-3
3 E=エディット	E	LOCATION	82	29	36	19	form-control mb-3/col-md-2
4 K=チェックボックス	F	SEL_FOLDER	139	29	95	18	col-md-2/form-check form-check-inline me-3
5 C=コンボボックス	G	FOLDER	247	29	117	18	form-select mb-3/col-md-3
6 P=プッシュボタン	H	PB1	380	29	72	21	btn btn-primary col-md-2/fa-search
7 U=サブフォーム		SF1	26	65	580	221	width: 100%;

ProgramTbl

変換ルール番号 2: サンプル CARD変換
 テーブル高さ 300px
 水平スクロールバー 表示しない

コントロール属性

コントロールタイプ	項目	コントロール名	x	y	幅	高さ	設定値
1 T=テーブル		Table27	12	13	772	302	
2 M=カラム		Column44	0	0	41	0	col-1
3 M=カラム		Column45	0	0	198	0	col-3
4 M=カラム		Column46	0	0	146	0	col-2
5 M=カラム		Column47	0	0	140	0	col-2
6 M=カラム		Column48	0	0	35	0	d-none
7 M=カラム		Column49	0	0	41	0	d-none
8 M=カラム		Column50	0	0	74	0	col-2
9 M=カラム		Column51	0	0	74	0	col-2
10 E=エディット	L	PNO	16	35	35	19	
11 E=エディット	M	PNAME	57	35	196	19	
12 E=エディット	N	PFOLD	255	35	138	19	
13 E=エディット	O	PPUB	401	35	132	19	
14 K=チェックボックス	P	PEXT	541	35	23	15	
15 K=チェックボックス	Q	POFL	576	35	23	15	
16 E=エディット	R	UPDATE	617	35	66	19	
17 E=エディット	S	UTIME	691	35	66	19	(付加情報) 付加情報で「要素の表示条件」を選択し、下記を設定 *ngIf="mq.getCustomProperty('UPDATE','IsEmpty',row.rowId)===''"

4) Program / ProgramTask / DataView / LogicView / FormList

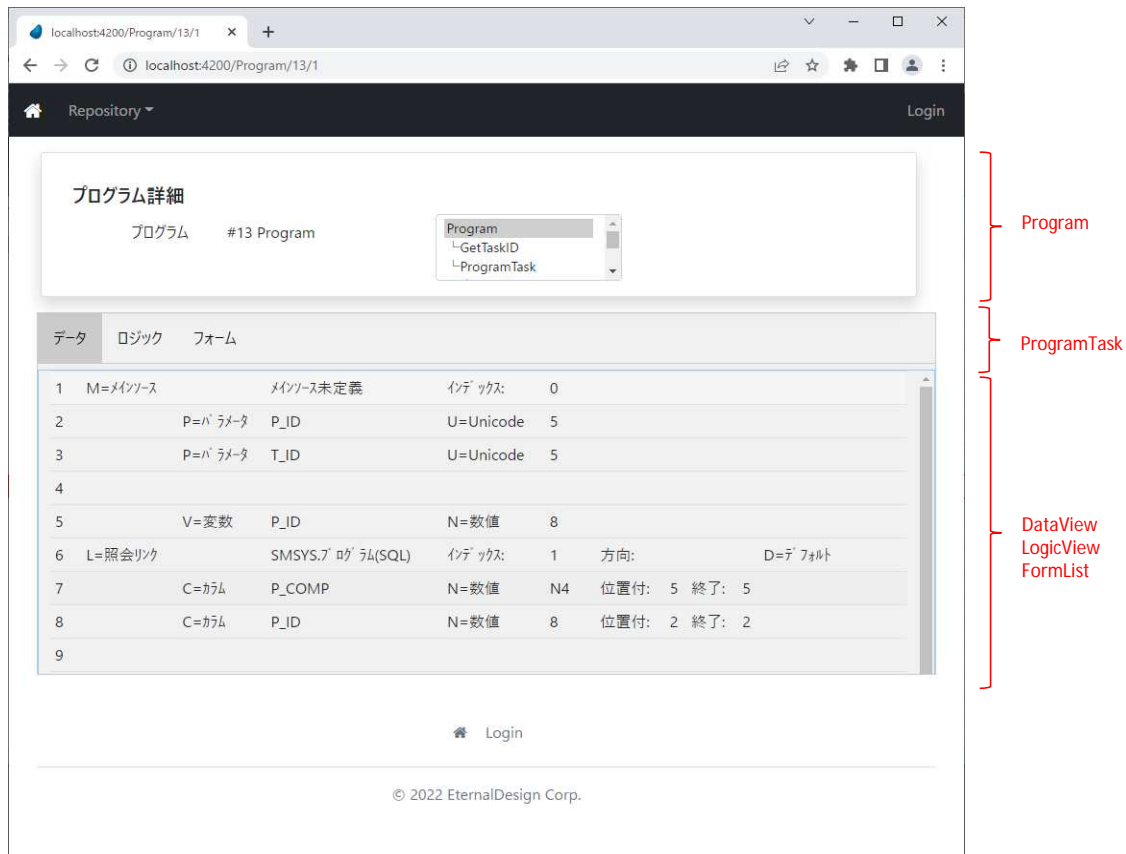


図2.5.3-4 プログラム詳細画面のデザイン例

Program

変換ルール番号 1: カードクラス適用(通常)
 CARD情報設定 タイトル: プログラム詳細
 エリア別ルール設定 エリア2を除外

コントロール属性

コントロールタイプ	項目	コントロール名	x	y	幅	高さ	設定値
1 O=グループ		Group54	12	3	551	48	width: 100%;/del_border
2 B=ラベル		LabelP	28	18	74	14	col-md-2 col-form-label text-md-end me-3
3 E=テキスト		PName	97	18	186	19	form-control mb-2/col-md-3
4 C=コンボボックス	O	Combo_box	289	18	167	18	form-select/col-md-3/size:3
5 U=サブフォーム		SF1	12	57	551	279	width: 100%;

ProgramTask

変換ルール番号 指定なし
 CSS直接記述 .tab_control { height: 380px;}

評価版では「CSS直接記述」ボタンが使えないのでCSSを直接編集して下さい

コントロール属性

コントロールタイプ	項目	コントロール名	x	y	幅	高さ	設定値
1 A=タブ	T	TAB	12	12	306	261	
2 U=サブフォーム		SF1	19	37	277	229	width: 100%;
3 U=サブフォーム		SF2	19	37	277	226	width: 100%;
4 U=サブフォーム		SF3	19	37	277	226	width: 100%;

DataView

変換ルール番号 1: サンプル TABLE変換
 ヘッダ 非表示

LogicView

変換ルール番号 1: サンプル TABLE変換
 ヘッダ 非表示

FormList

変換ルール番号 1: サンプル TABLE変換

5) DataList / DataTbl

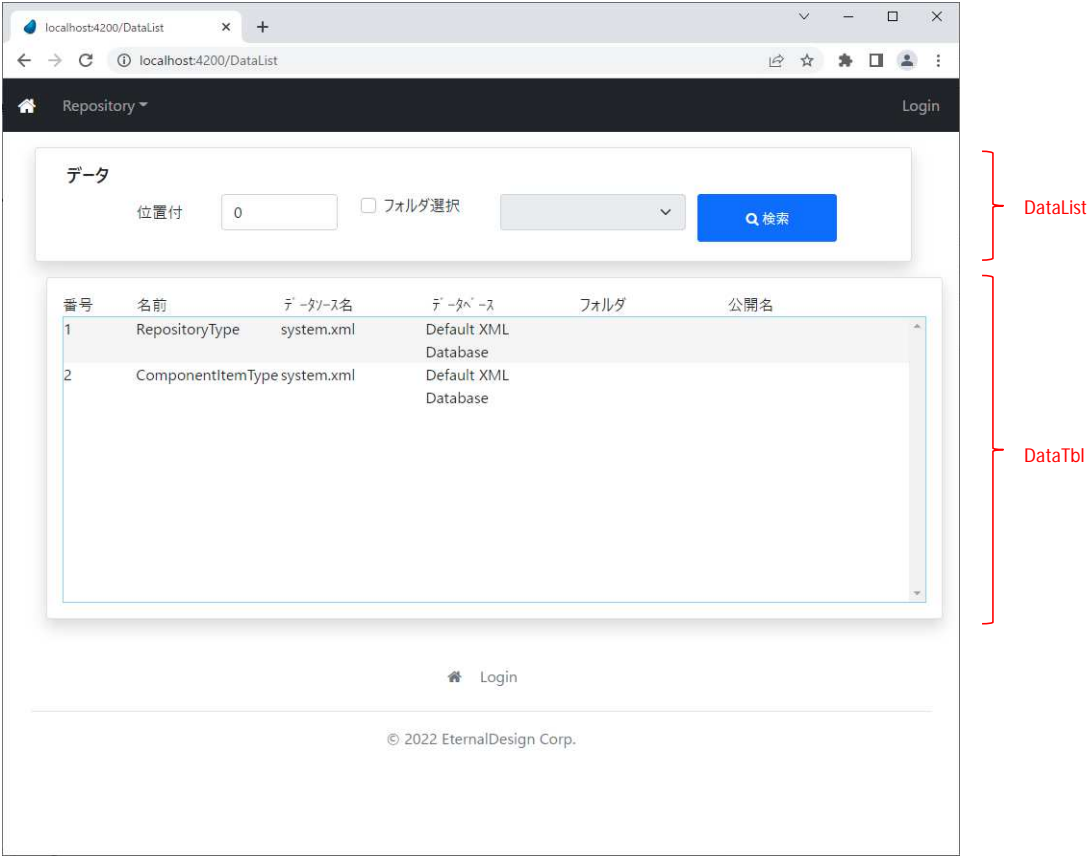


図2.5.3-5 データー一覧画面のデザイン例

DataList

変換ルール番号 1: カードクラス適用(通常)
CARD情報設定 タイトル: データ
エリア別ルール設定 エリア2を除外

コントロール属性 「ProgramList」からコピー

コントロールタイプ	項目	コントロール名	x	y	幅	高さ	設定値
1 O=グループ		Group7	12	12	470	47	del border
2 B=ラベル		Label2	26	29	50	14	col-md-2 col-form-label text-md-end me-3
3 E=エディット	E	LOCATION	82	29	36	19	form-control mb-3/col-md-2
4 K=チェックボックス	F	SEL_FOLDER	139	29	95	18	col-md-2/form-check form-check-inline me-3
5 C=コンボボックス	G	FOLDER	247	29	117	18	form-select mb-3/col-md-3
6 P=プッシュボタン	I	PB1	380	29	72	21	btn btn-primary col-md-2/fa-search
7 U=サブフォーム		SF1	26	65	580	221	width: 100%;

DataTbl

変換ルール番号 2: サンプル CARD変換
テーブル高さ 300px
水平スクロールバー 表示しない

6) DataDetail / DataCol / DataIndex

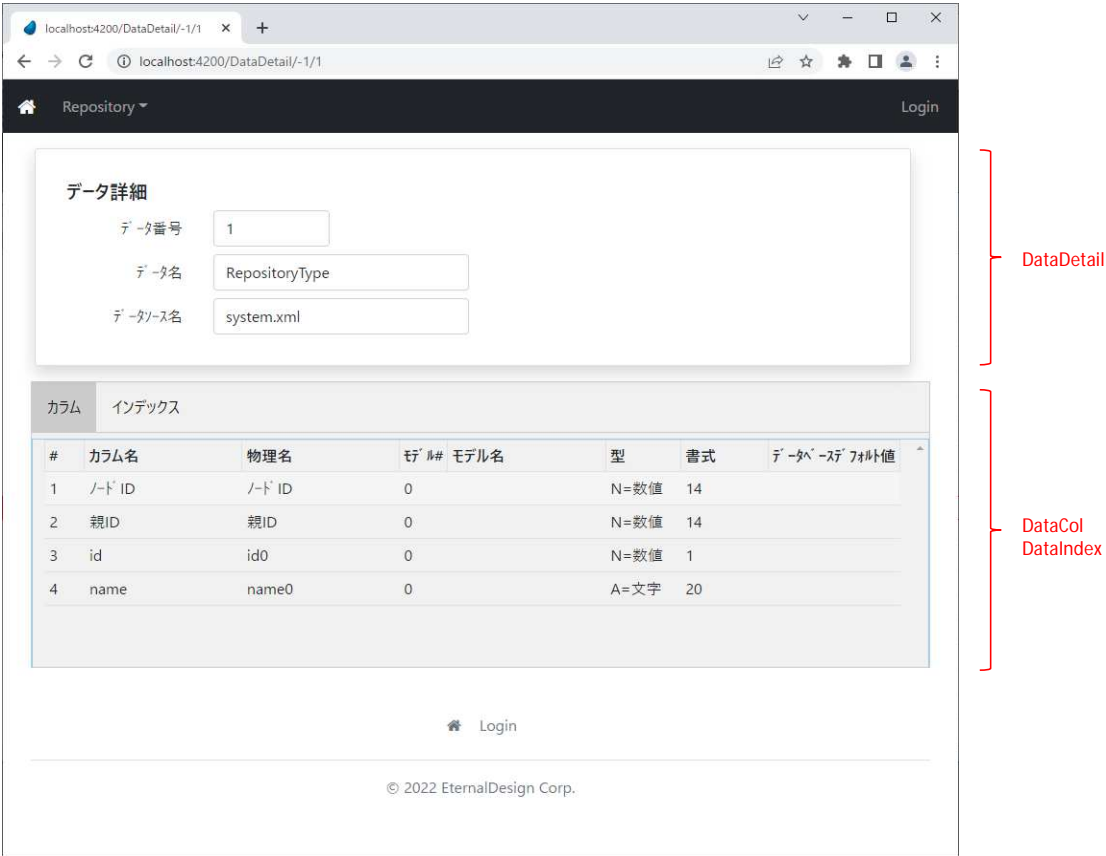


図2.5.3-6 データ詳細画面のデザイン例

DataDetail

変換ルール番号 1: カードクラス適用(通常)
CARD情報設定 タイトル: データ詳細
CSS直接記述 .tab_control { height: 300px; }
評価版では「CSS直接記述」ボタンが使えないのでCSSを直接編集して下さい

コントロール属性

コントロールタイプ	項目	コントロール名	x	y	幅	高さ	設定値
1 O=グループ		Group4	10	12	424	96	pb-2/del border
2 B=ラベル		Label6	19	31	50	14	col-md-2 col-form-label text-md-end me-2
3 E=エディット	K	DNo	79	31	54	19	form-control mb-2/col-md-2
4 B=ラベル		Label8	19	49	38	14	col-md-2 col-form-label text-md-end me-2
5 E=エディット	L	DName	79	49	192	19	form-control mb-2/col-md-4
6 B=ラベル		Label10	17	67	56	14	col-md-2 col-form-label text-md-end me-2
7 E=エディット	M	DTName	79	67	192	19	form-control/col-md-4
8 A=タブ	N	Tab1	12	114	426	270	
9 U=サブフォーム		SF1	19	139	412	238	width: 100%;
10 U=サブフォーム		SF2	22	139	412	238	width: 100%;

DataCol

変換ルール番号 1: サンプル TABLE変換

DataIndex

変換ルール番号 1: サンプル TABLE変換

7) ModelList / ModelTbl

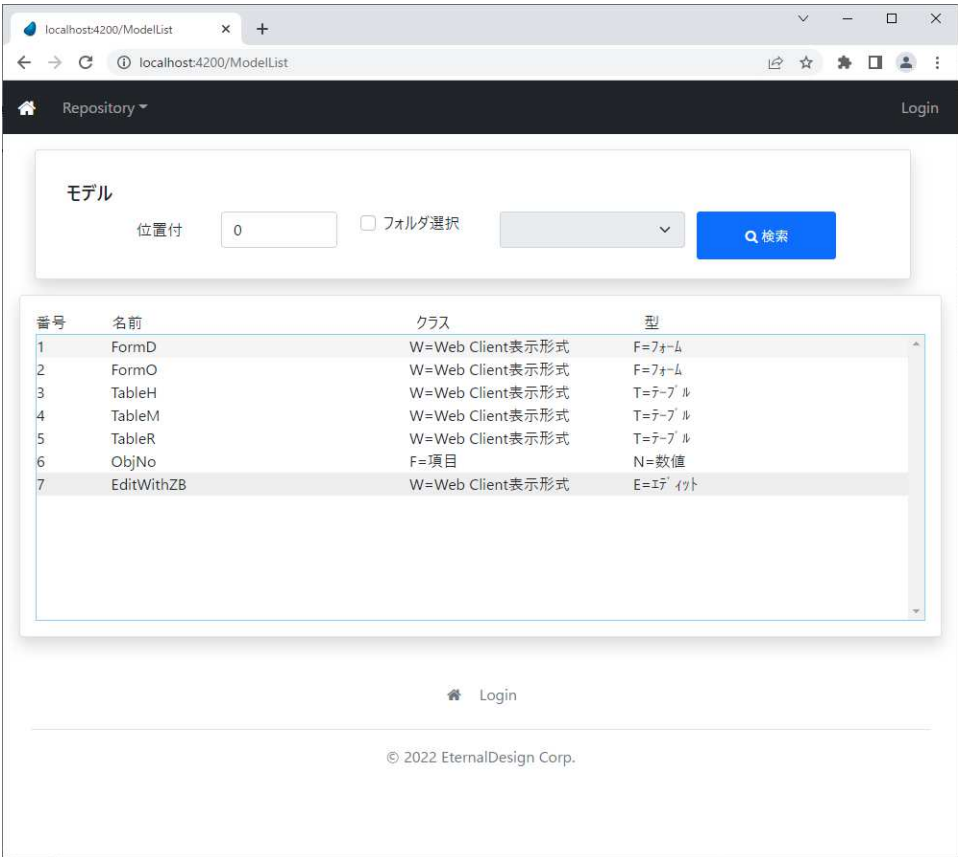


図2.5.3-7 モデルー覧画面のデザイン例

ModelList

変換ルール番号 1: カードクラス適用(通常)
CARD情報設定 タイトル: モデル
エリア別ルール設定 エリア2を除外

コントロール属性 「ProgramList」からコピー

	コントロールタイプ	項目	コントロール名	x	y	幅	高さ	設定値
1	O=グループ		Group7	12	12	470	47	del border
2	B=ラベル		Label2	26	29	50	14	col-md-2 col-form-label text-md-end me-3
3	E=エディット	E	LOCATION	82	29	36	19	form-control mb-3/col-md-2
4	K=チェックボックス	F	SEL_FOLDER	139	29	95	18	col-md-2/form-check form-check-inline me-3
5	C=コンボボックス	G	FOLDER	247	29	117	18	form-select mb-3/col-md-3
6	P=プッシュボタン	I	PB1	380	29	72	21	btn btn-primary col-md-2/fa-search
7	U=サブフォーム		SF1	26	65	580	221	width: 100%;

ModelTbl

変換ルール番号 2: サンプル CARD変換
テーブル高さ 300px
水平スクロールバー 表示しない

8) ZoomButtonTest

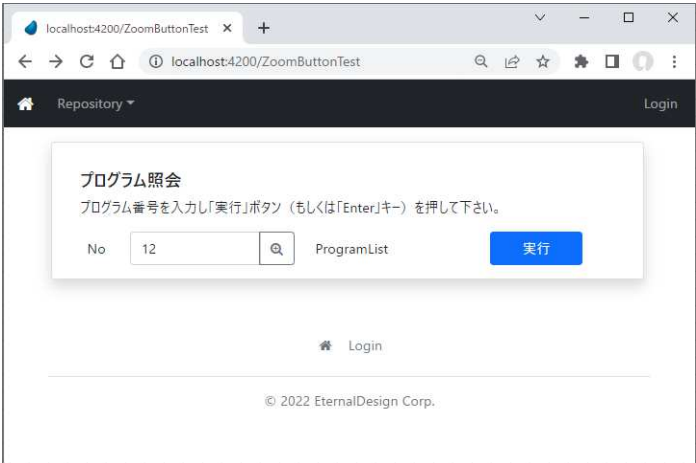


図2.5.3-8a ズームボタンテスト画面のデザイン例 (幅768px以上)

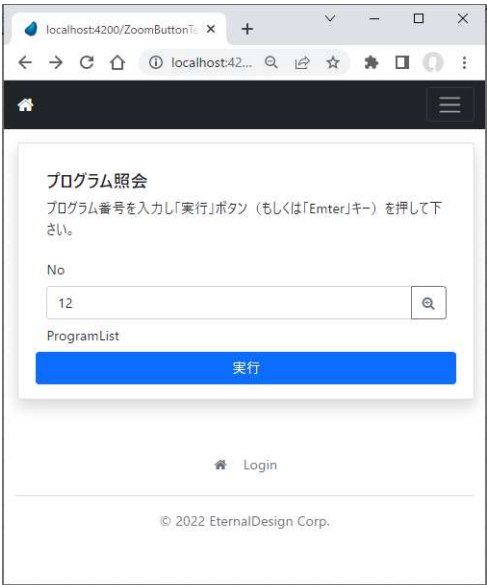


図2.5.3-8b ズームボタンテスト画面のデザイン例 (幅768px未満)

変換ルール番号 1: カードクラス適用(通常)
CARD情報設定 タイトル: プログラム照会
説明文: プログラム番号を入力し「実行」ボタン(もしくは「Enter」キー)を押して下さい。

コントロール属性								設定値
コントロールタイプ	項目	コントロール名	x	y	幅	高さ		
1 O=グループ		Group4	12	12	605	108	del border	
2 B=ラベル		Label2	35	34	24	23	col-md-1 col-form-label text-md-end me-1	
3 E=エディット	E	V PNO	67	34	89	23	form-control/fa-search-plus/col-md-4/zoombtn:btn-outline-secondary	
4 E=エディット	G	PName	162	34	308	23	form-control/col-md-4	
5 P=プッシュボタン	L	PB1	476	34	72	23	btn btn-primary col-md-2	

9) SelectProgram

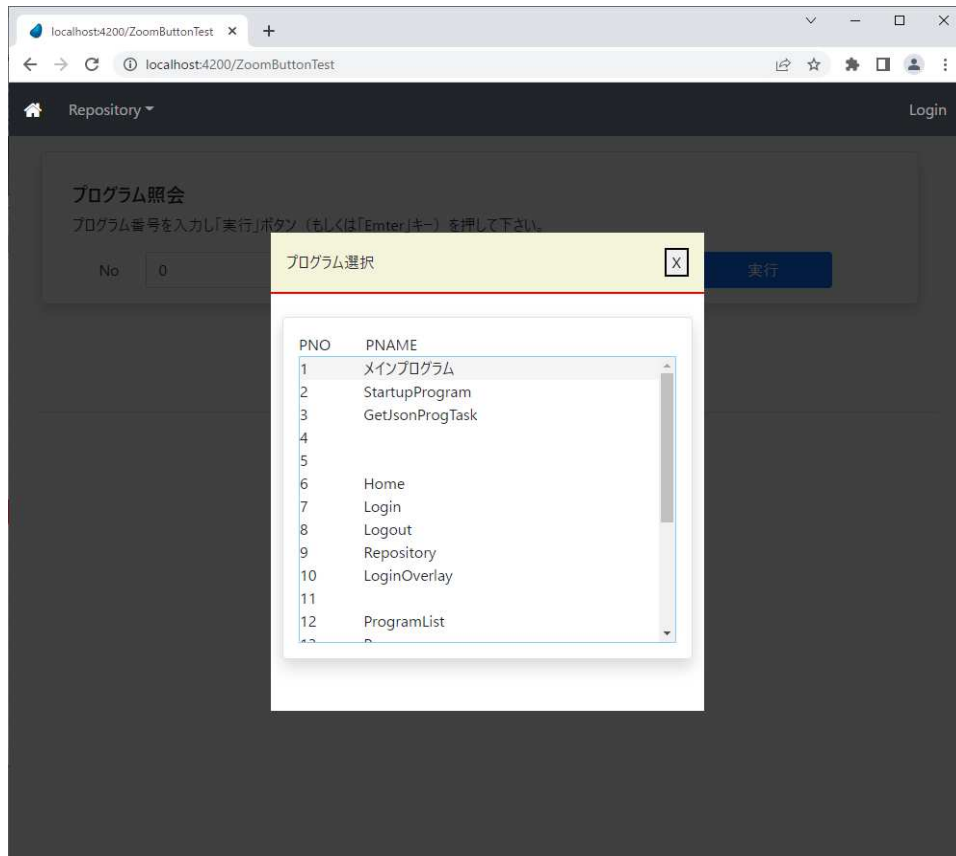


図2.5.3-9 プログラム選択画面のデザイン例

変換ルール番号 8 : サンプル CARD変換(オーバーレイ用)
 テーブル高さ 300px
 水平スクロールバー 表示しない
 TSファイルの定義 モーダルウィンドウ幅: 450px
 モーダルウィンドウ高さ: 500px
 モーダルウィンドウのタイトル: プログラム選択

10) GridTest

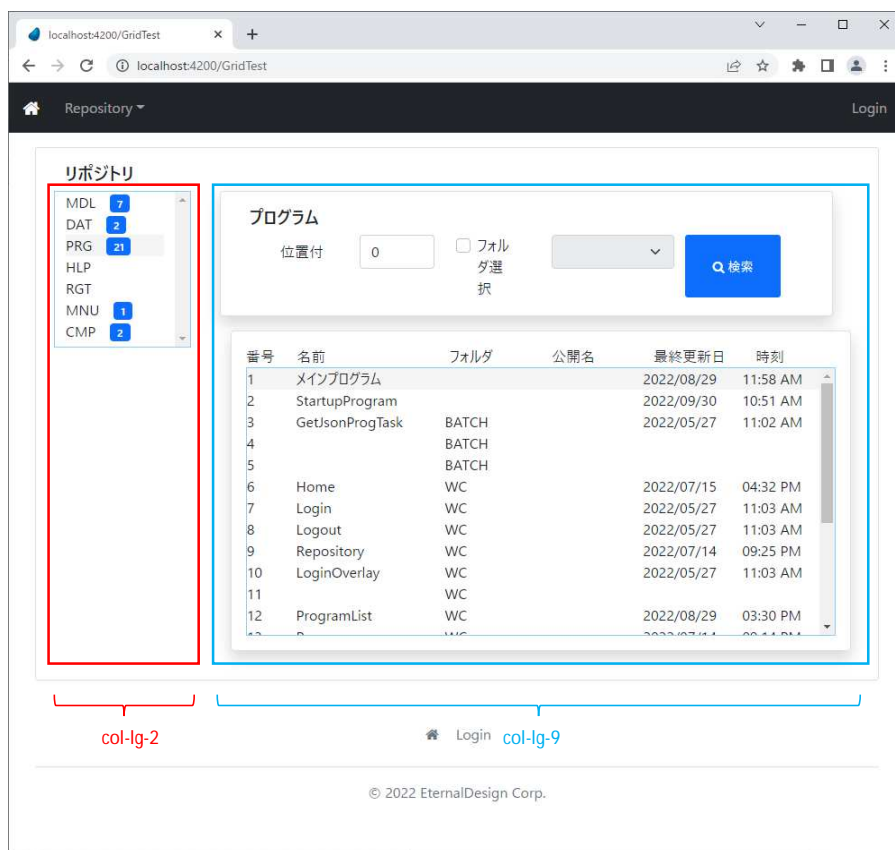


図2.5.3-10a グリッドテスト画面のデザイン例 (幅992px以上)



図2.5.3-10b グリッドテスト画面のデザイン例 (幅992px未満)

変換ルール番号 7: サンプル CARD変換(グリッド利用)
 CARD情報設定 タイトル: リポジトリ
 テーブル高さ 200px
 水平スクロールバー 表示しない

コントロール属性

コントロールタイプ	項目	コントロール名	x	y	幅	高さ	設定値
1 T=テーブル		Tbl	22	28	29	303	order-last order-lq-first me-3/d-block col col-lq-2
2 M=カラム		Col1	0	0	23	0	d-block col-1 d-lq-none
3 M=カラム		Col2	0	0	134	0	d-block col-5 d-lq-none
4 M=カラム		Col3	0	0	46	0	d-block col-1 d-lq-none
5 M=カラム		Col4	0	0	94	0	d-none d-lq-block col-lq
6 E=エディット	E	did	26	50	14	23	
7 E=エディット	F	name	49	50	126	23	
8 E=エディット	D	CNT	183	50	38	23	
9 E=エディット	G	shortname	229	50	86	23	(付加情報) 付加情報で「構文追加(後置)」を選択し、下記を設定 {{mg.getCustomProperty('shortname','count',row.rowId)}}
10 U=サブフォーム		SF1	497	28	319	303	mb-3 d-block col-lq-9

11) DrawChart

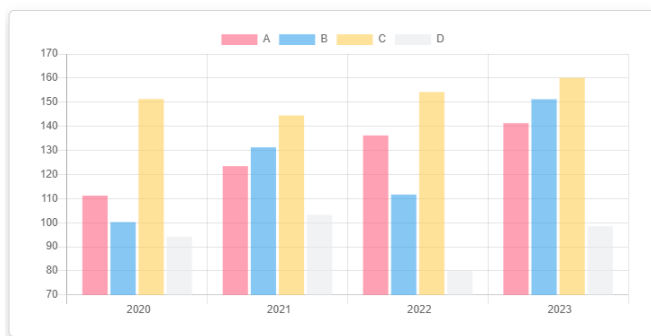


図2.5.3-11 チャート描画面面のデザイン例

変換ルール番号 5:チャート表示
 変数情報設定 chartType 2=値 bar
 datasets 1=コントロール P_ChartData
 labels 1=コントロール P_Label

コントロール属性

コントロールタイプ	項目	コントロール名	x	y	幅	高さ	設定値
1 O=グループ		Group1	12	12	352	268	d:none
2 B=ラベル		Label3	34	40	56	14	
3 E=エディット	C	P_ChartData	96	40	96	18	
4 B=ラベル		Label5	34	64	56	14	
5 E=エディット	D	P_Label	96	64	96	18	

モジュール設定ウィザードにより「Chart.js」のインストールが必要になります。

12) GraphTest

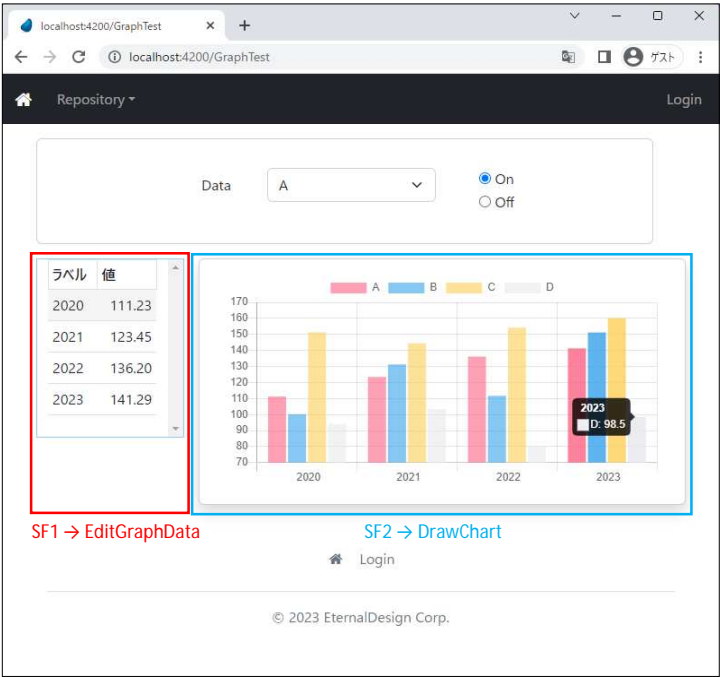


図2.5.3-12 グラフテスト画面のデザイン例

GraphTest

変換ルール番号 1: カードクラス適用(通常)
エリア別ルール設定 エリア2を除外

コントロール属性

	コントロールタイプ	項目	コントロール名	x	y	幅	高さ	設定値
1	O=グループ		Group18	12	12	627	111	mb-3/del_border
2	B=ラベル		Label1(Data)	36	42	62	20	col-form-label col text-end me-3
3	C=コンボボックス	C	DataNo	116	42	268	18	form-select me-3/col/size:1
4	D=ラジオボタン	G	V Radio	407	42	61	62	col/form-check form-check-inline/me-3
5	U=サブフォーム		SF1	12	162	474	228	me-3 d-block col mb-3
6	U=サブフォーム		SF2	507	162	454	228	d-block col

EditGraphData

変換ルール番号 1: サンプル TABLE変換
テーブル高さ 200px
水平スクロールバー 表示しない

コントロール属性

	コントロールタイプ	項目	コントロール名	x	y	幅	高さ	設定値
1	T=テーブル		Table1	12	12	318	244	
2	M=カラム		Column1	0	0	50	0	d-none
3	M=カラム		Column2	0	0	44	0	d-none
4	M=カラム		Column4	0	0	74	0	
5	M=カラム		Column6	0	0	98	0	
6	P=プッシュボタン		PB Edit	16	34	36	21	btn btn-secondary/fa-pencil
7	E=エディット	O	XNo	66	34	36	19	pull-right
8	E=エディット	R	LabelX	110	34	66	19	
9	E=エディット	Q	Val	184	34	90	19	pull-right
10	P=プッシュボタン		PB Update	16	59	36	21	btn btn-danger/fa-check
11	P=プッシュボタン		PB Cancel	16	84	36	21	btn btn-warning/fa-undo

参考) StartupProgram

ルーティング情報で「Navbarを作成しない」にチェックを入れたときのテンプレートの変換例です。

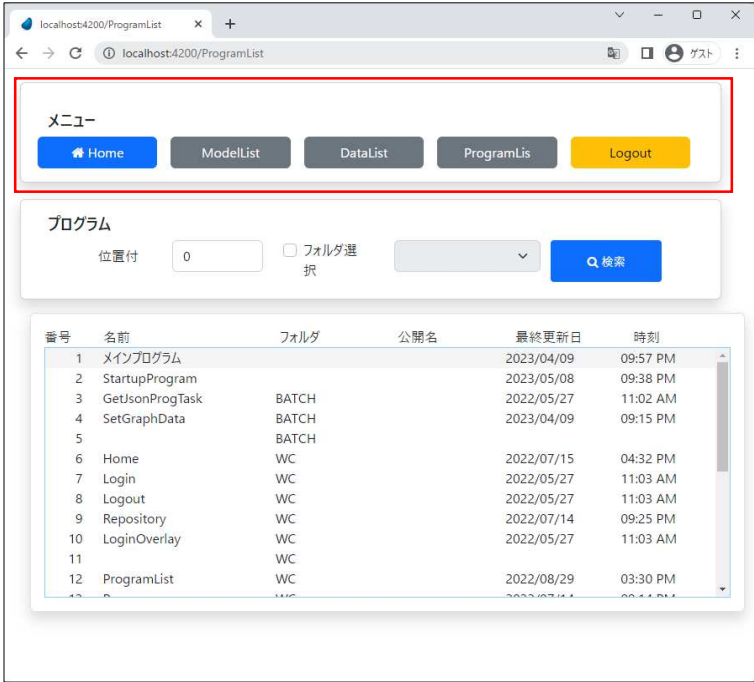


図2.5.3-13 スタートアッププログラム画面のデザイン例

変換ルール番号 1:カードクラス適用(通常)
CARD情報設定 タイトル:メニュー
エリア別ルール設定 エリア2を除外

コントロール属性									
コントロールタイプ	項目	コントロール名	x	y	幅	高さ	設定値		
1 O=グループ		GRP1	12	12	415	95	del border		
2 P=フッシュボタン	C	PB1	19	31	72	21	btn btn-primary col me-3/fa-home		
3 P=フッシュボタン	D	PB2	98	30	72	21	btn btn-secondary col me-3		
4 P=フッシュボタン	E	PB3	177	29	72	21	btn btn-secondary col me-3		
5 P=フッシュボタン	F	PB4	256	29	72	21	btn btn-secondary col me-3		
6 P=フッシュボタン	G	PB5	335	29	72	21	btn btn-warning col me-3		
7 U=サブフォーム		SF1	12	113	415	202	mt-1		

< Navbar作成時に「サンプル1 / normal」以外を選択していた場合の対処方法 >

Navbar作成時に「サンプル1 / normal」以外のテンプレートを選択していた場合は、ブラウザで表示したとき、画面の上部等に余白が表示されます。その場合は下記のように対処してください。

Visual Studio Codeでindex.htmlを開きます
'body, タグの属性 'style' を除去します
ファイルを保存します



楽々ウェブクライアント / SMSYS Ver 1.01新機能の紹介

1.ズームボタン(ShowZoomButton)に対応

MAGIC xpa Ver 4.8で利用できるようになったエディットのズームボタン(ShowZoomButton)に対応しました。エディットの「ズームボタンを表示」プロパティが有効な場合は、Bootstrapの「input-group」クラスを使用し、ボタンをテキスト入力と一体化させます。更にGUI編集画面からボタンのデザインやアイコンを指定することも可能にしました。

サンプルプログラムのフォーム

エディットの「ズームボタンを表示」プロパティが有効な場合は、フォームエディタ上にズーム用のボタンを表示します。

サンプルプログラム「ZoomButtonTest」ではエディット項目のスタイルでモデルを使用しています。

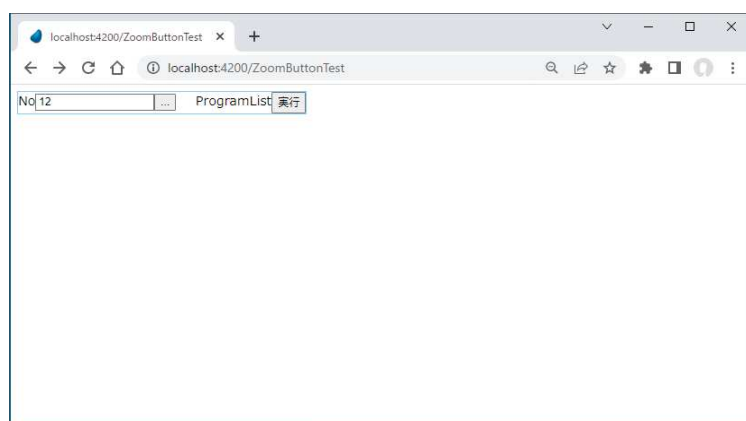
ズームボタンを表示する方法には幾つかの方法があります。

- フォームエディタ上でコントロールのプロパティを直接指定する
- フォームエディタ上でモデル(ズームボタンを表示)を指定する
- データ項目の該当のスタイルでプロパティを直接指定する
- データ項目の該当のスタイルでモデル(ズームボタンを表示)を指定する

楽々ウェブクライアントでは上記の何れのパターンにも対応しました。

変換処理について

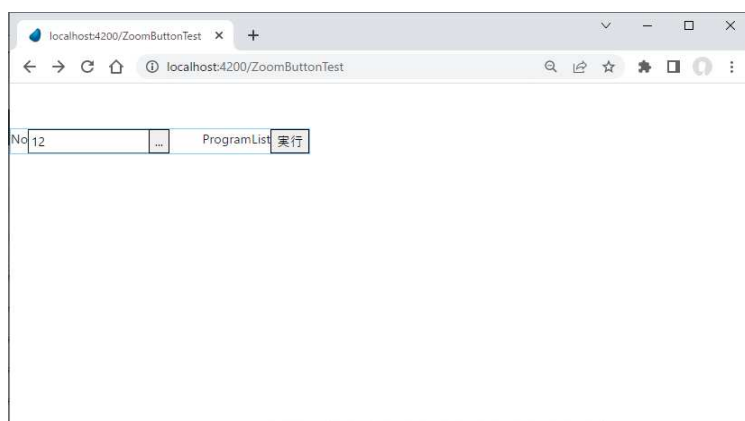
そのままサンプルを実行した場合のブラウザ表示結果は次のようになります。



楽々ウェブクライアントで何も指定せずに変換してみます。

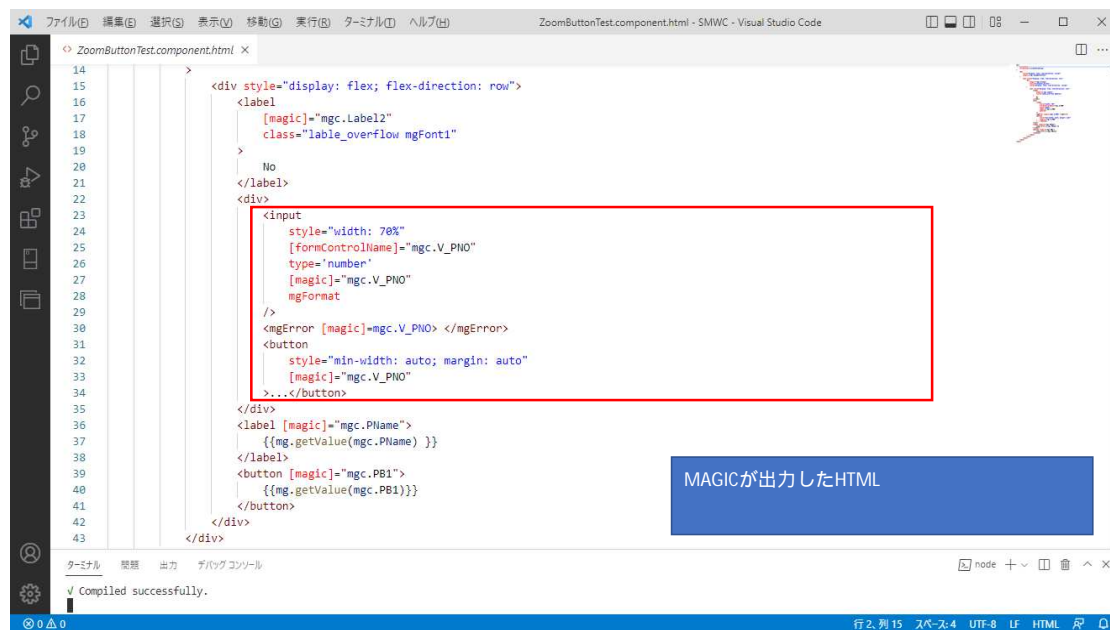
変換ルール番号や、フォーム確認からのコントロール属性を入力せずに変換します。

変換後の実行結果は次の様になりました。



ブラウザの表示結果だけでは、さして大きな違いは見られないように思えます。

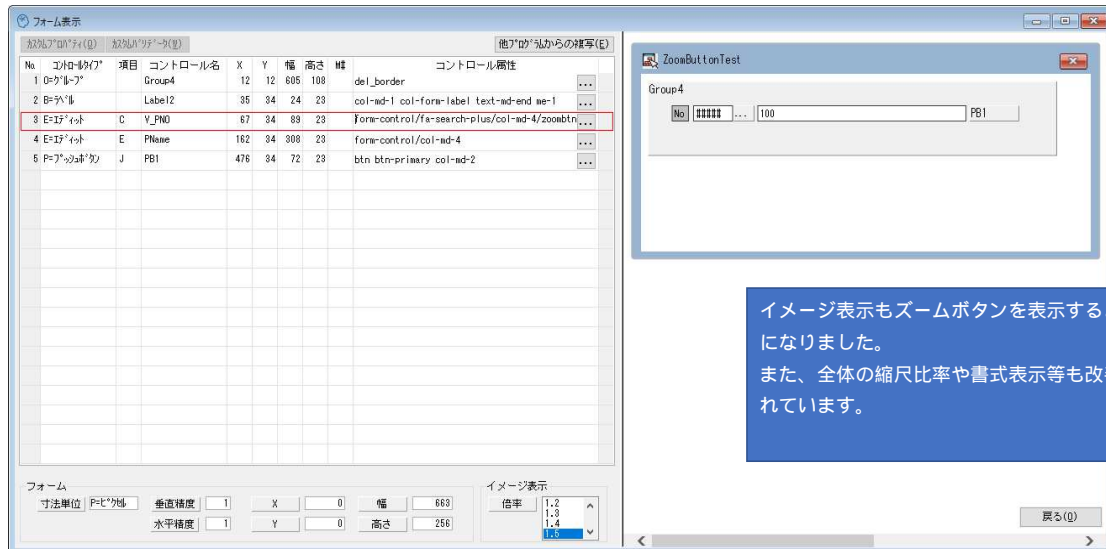
HTMLを比較してみます。



このようにして、楽々ウェブクライアントでは、bootstrapのInput group(インプットグループ)クラスを使用してズームボタンを実現します。これにより、テキスト入力とボタンが一体として配置されるようになります。

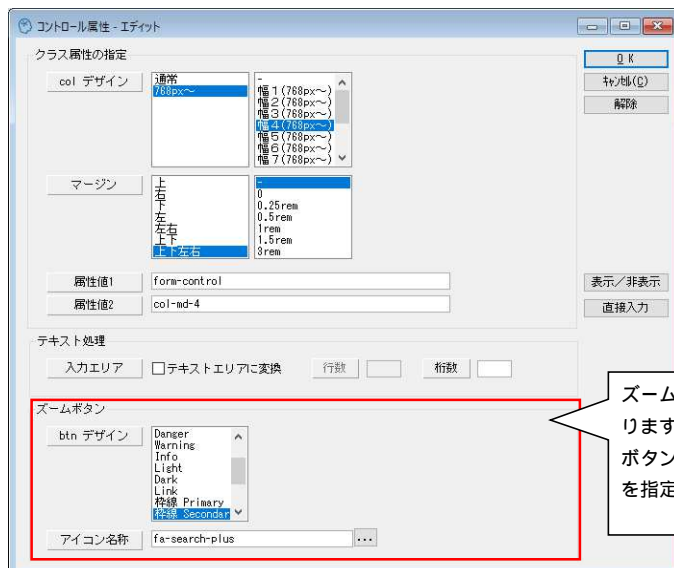
ボタンのデザイン変更など

フォームやコントロールに対してデザインを施した例を見えます。



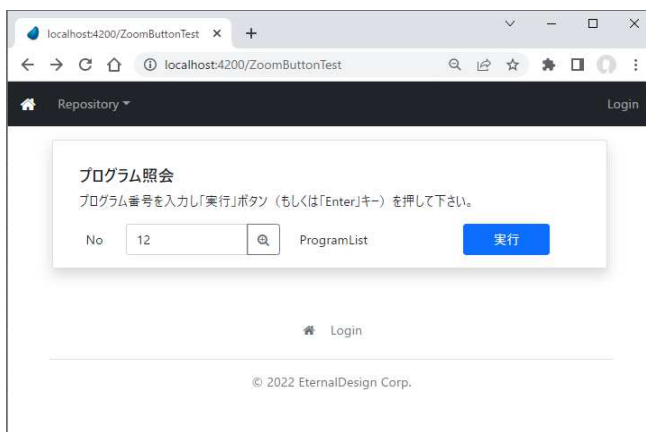
イメージ表示もズームボタンを表示するようになりました。
また、全体の縮尺比率や書式表示等も改善されています。

コントロール属性設定画面も拡張されました。
ボタンのデザイン等も簡単に指定できるようになりました。

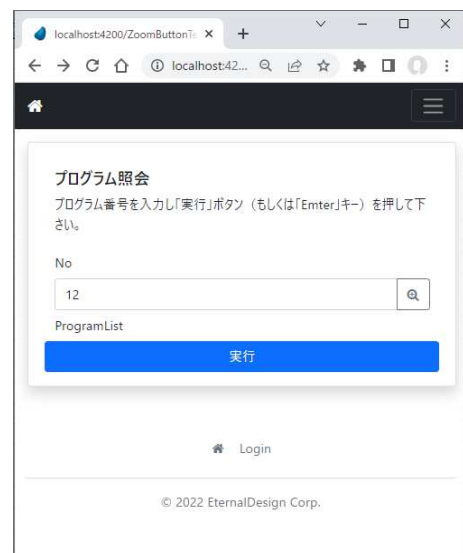


ズームボタンが存在する場合は、このエリアが有効になります。
ボタンのデザインや「...」を置き換えるためのアイコンを指定することが可能です。

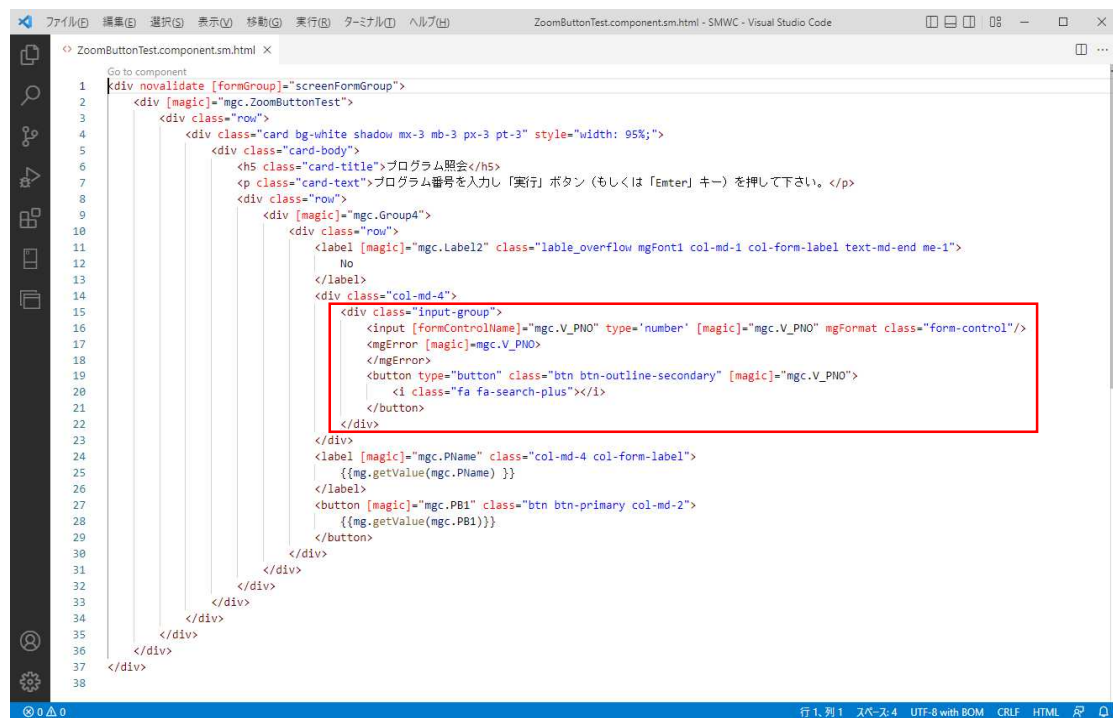
変換例を示します。



Bootstrapのグリッドシステムを使用したフレキシブルなレイアウトの中で、テキスト入力と一体化したズームボタンを実現することができました。



HTML変換結果



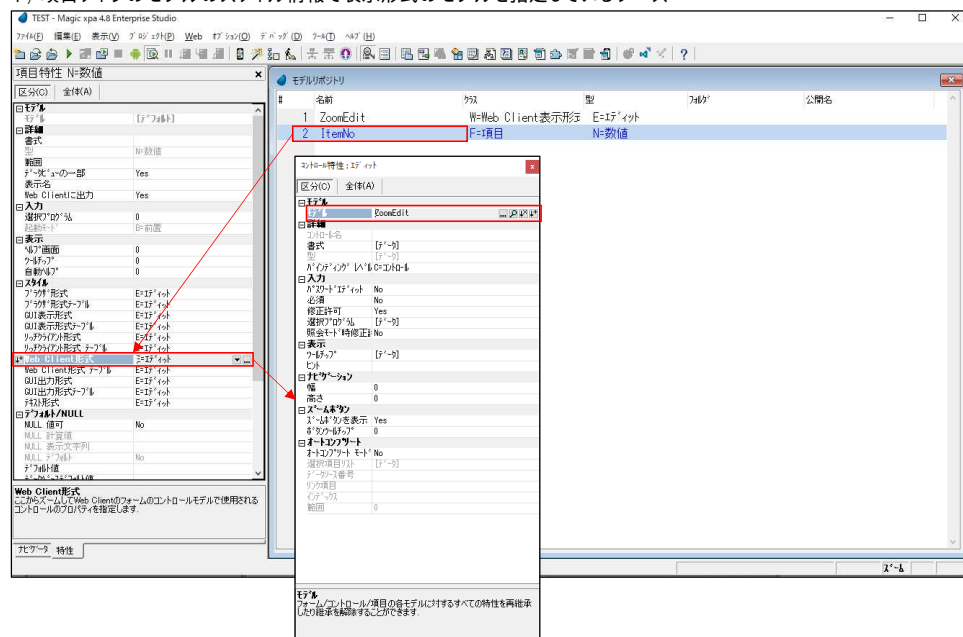
```
1 <div novalidate [formGroup]="screenFormGroup">
2   <div [magic]="mgc.ZoomButtonTest">
3     <div class="row">
4       <div class="card bg-white shadow mx-3 mb-3 px-3 pt-3" style="width: 95%;">
5         <div class="card-body">
6           <h5 class="card-title">プログラム照会</h5>
7           <p class="card-text">プログラム番号を入力し「実行」ボタン（もしくは「Enter」キー）を押して下さい。</p>
8           <div class="row">
9             <div [magic]="mgc.Group4">
10              <div class="row">
11                <label [magic]="mgc.Label2" class="lable_overflow mgFont1 col-md-1 col-form-label text-md-end me-1">
12                  No
13                </label>
14                <div class="col-md-4">
15                  <div class="input-group">
16                    <input [formControlName]="mgc.V_PNO" type="number" [magic]="mgc.V_PNO" mgFormat class="form-control"/>
17                    <mgError [magic]="mgc.V_PNO">
18                      </mgError>
19                    <button type="button" class="btn btn-outline-secondary" [magic]="mgc.V_PNO">
20                      <i class="fa fa-search-plus"></i>
21                    </button>
22                  </div>
23                </div>
24                <label [magic]="mgc.PName" class="col-md-4 col-form-label">
25                  {{mg.getValue(mgc.PName)}}
26                </label>
27                <button [magic]="mgc.PB1" class="btn btn-primary col-md-2">
28                  {{mg.getValue(mgc.PB1)}}
29                </button>
30              </div>
31            </div>
32          </div>
33        </div>
34      </div>
35    </div>
36  </div>
37</div>
```

2. スタイル情報に使用されたモデルのリファレンス

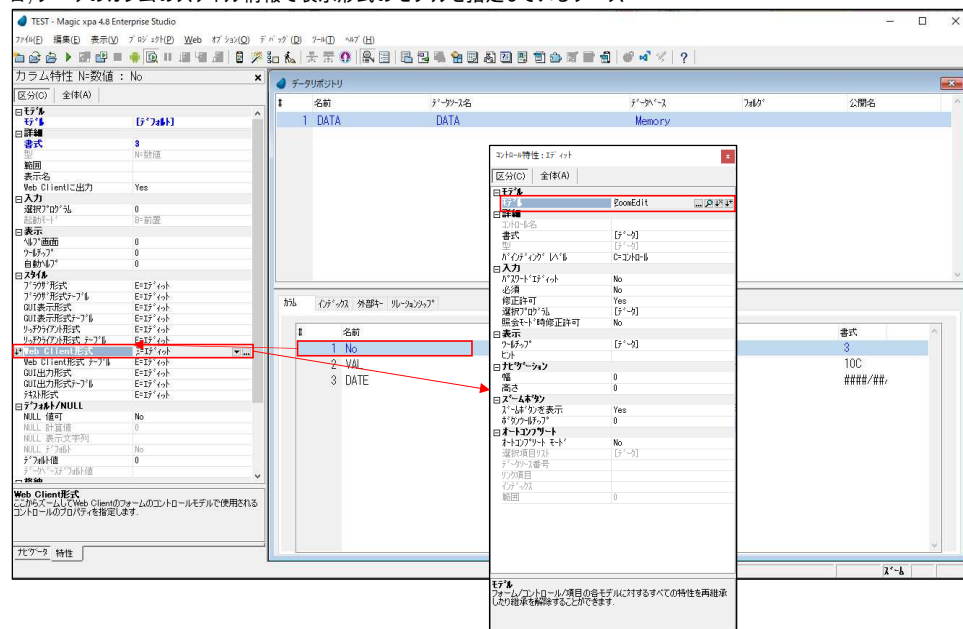
モデルのリファレンスを、スタイル情報(モデル、データ、プログラム項目)で使用しているモデルに対しても取得するようにしました。
 但し、現状は「GUI表示形式」「GUI表示形式テーブル」「リッチクライアント表示形式」「リッチクライアント表示形式テーブル」「WebClient形式」「WebClient形式テーブル」のみに限定しています。
 これらのスタイル情報にモデルの利用を取得したときは、「モデル」一覧画面のクロスリファレンスボタン「...」から確認できるようになりました。

<検証データ>

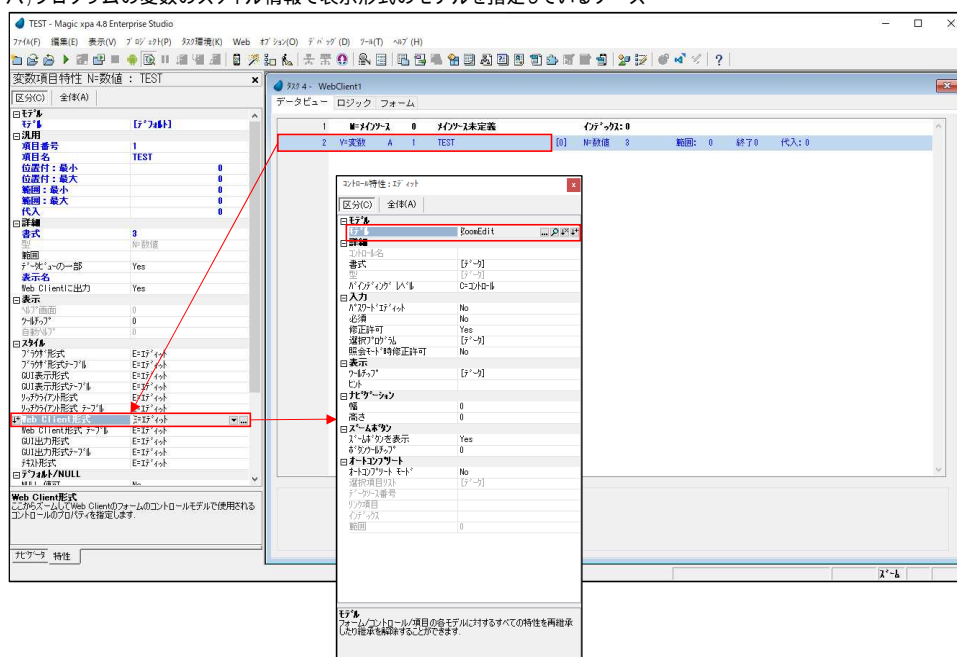
イ) 項目タイプのモデルのスタイル情報で表示形式のモデルを指定しているケース



ロ) データのカラムのスタイル情報で表示形式のモデルを指定しているケース

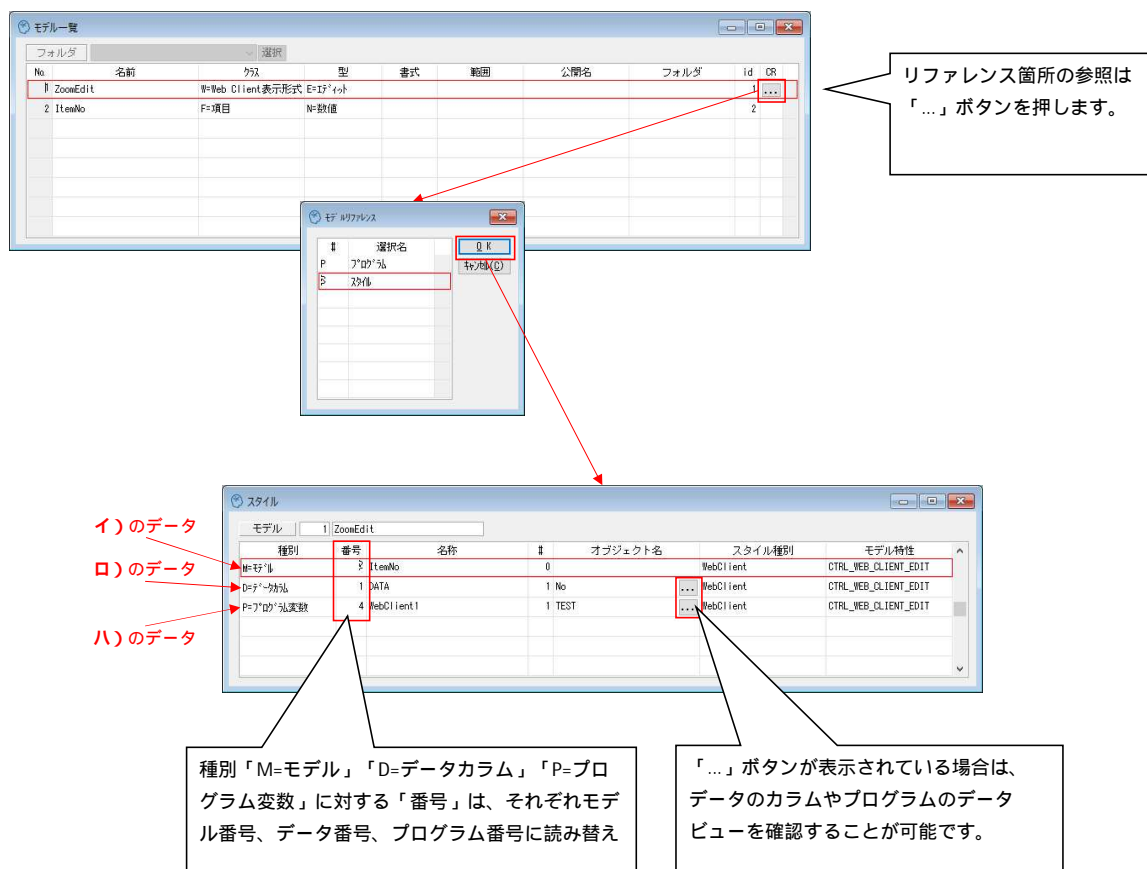


ハ) プログラムの変数のスタイル情報で表示形式のモデルを指定しているケース



<リファレンスの確認方法>

モデル一覧プログラムを起動します。
「CR」カラムのリファレンスボタン(関連がある場合に表示)を押します。
選択ポップアップが出た場合は「S:スタイル」を選択します。



楽々ウェブクライアント / SMSYS Ver 1.02新機能の紹介

1. グリッドシステム管理機能の強化

以前のバージョンまでは、A) Webスタイルを「繰り返し可能な項目」で設定したテーブルのカラム間でのグリッド指定(CARDタイプを指定)や、グループに関連付けたコントロール(ラベル、エディット、ボタン、コンボボックス、チェックボックス、ラジオボタン)間でのグリッド指定に留まっていましたが、今回のバージョンでは、テーブル、サブフォーム、グループ、タブ等のコントロール間でのグリッド指定ができるようになりました。これにより、ページを構成するより大きな単位でのレイアウトを管理することができるようになりました。

以前のバージョンまでのグリッド定義機能

復習がてら、以前のバージョンで利用できるグリッド定義機能の実例を紹介します。

1) テーブルのカラム間で幅を決める

楽々ウェブクライアントでは、テーブルのカラムにグリッドシステムを適用させることが可能です。例えば、下図のようなテーブルを持つプログラムを変換してみます。

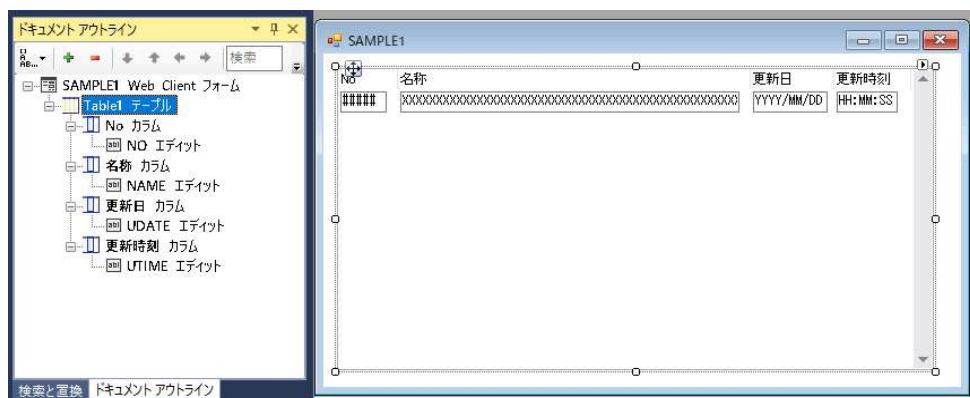
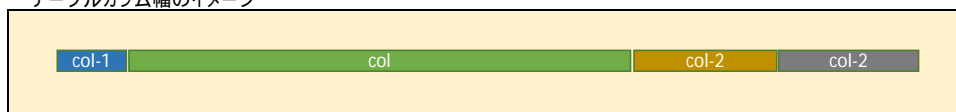


図1 テーブルをもつプログラムのMAGICフォームエディタ画面

「No」「名称」「更新日」「更新時刻」のカラムにそれぞれ、「col-1」「col」「col-2」「col-2」のクラスをセットすることにより「1:7:2:2」の比率で表示します。数字のつかない「col」は等幅で、残りのカラム(この場合は12 - 1 - 2 - 2 = 7)を等分します。

テーブルカラム幅のイメージ



楽々ウェブクライアントでデザインした結果は次のようになります。実際にカラム幅が指定した配分になっていることを確認することができます。

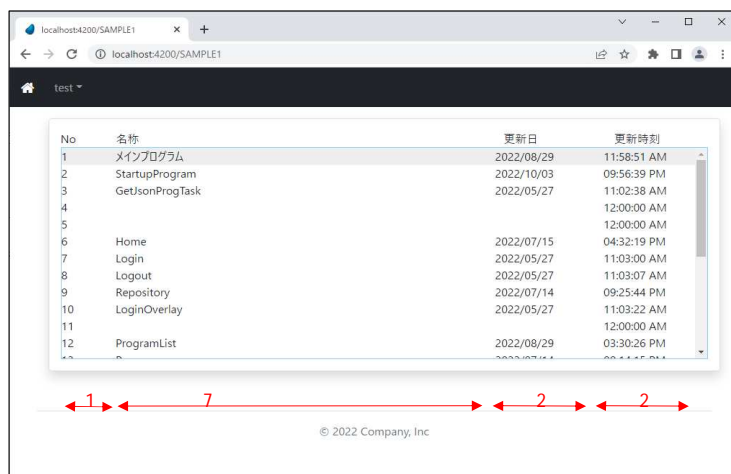


図2 楽々ウェブクライアントデザイン例(テーブルのカラム)

2) ラベルやエディット等のコントロールをグループでまとめ間隔を定義

次に、テーブルを持たないフォームの場合です。

楽々ウェブクライアントでは複数のコントロールをグループコントロールで括ることにより、配置をコントロールすることが可能です。

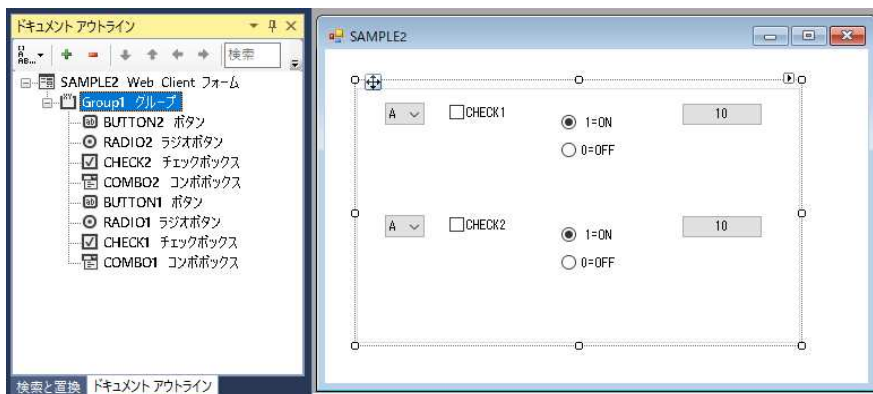
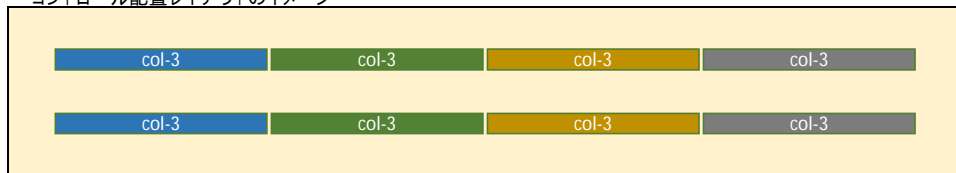


図3 グループを利用したプログラムのMAGICフォームエディタ画面

コントロール「COMBO1」「CHECK1」「RADIO1」「BUTTON1」にそれぞれ「col-3」のクラス属性をセットすることにより「1:1:1:1」の比率で表示させます。コントロール「COMBO2」「CHECK2」「RADIO2」「BUTTON2」も同様に設定し均等にコントロールを配置してみます。
(全て同じ幅であれば「col-3」ではなく「col」でもOKです)

コントロール配置レイアウトのイメージ



楽々ウェブクライアントでデザインした結果は次のようになります。

実際にコントロールの間隔が当分になっていることを確認することができます。

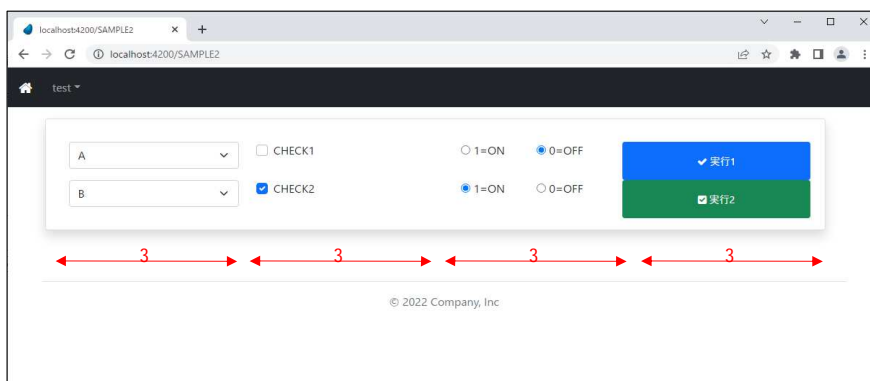


図4 楽々ウェブクライアントデザイン例(グループによる配置)

テーブルやサブフォーム間でのレイアウト機能

今回のバージョンでは、例えば、下図のように、テーブルの右や下段にサブフォームがあるようなケースでは、テーブルとサブフォーム、サブフォームとサブフォームの画面表示割合を設定することができませんでした。
これを、コントロール属性で設定することにより実現できるようになっています。

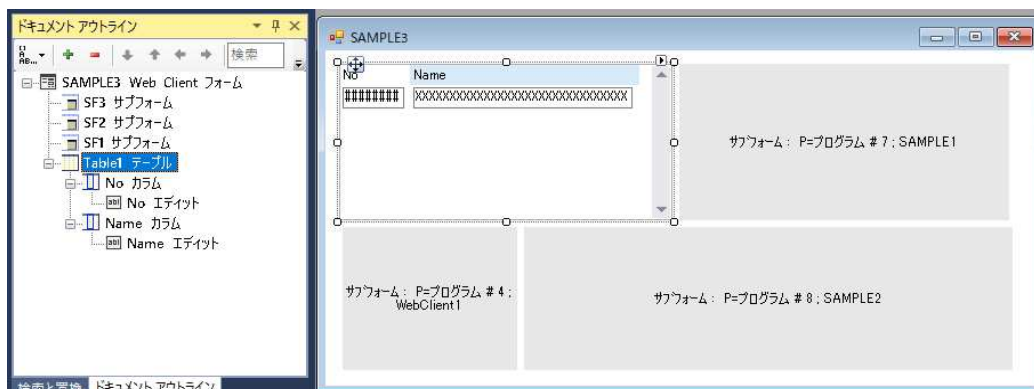
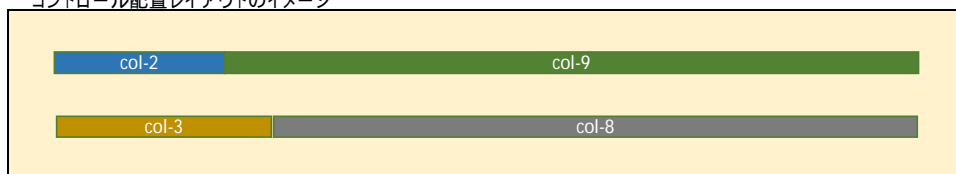


図5 テーブルと幾つかのサブフォームが混在するプログラムのMAGICフォームエディタ画面

例えば、コントロール「Table1」「SF1」「SF2」「SF3」にそれぞれ「col-2」「col-10」「col-3」「col-9」のクラス属性をセットすることにより「1:1」「1:2」の比率で表示させることができます。
(実際は、マージン等を考慮するため、「col-10」「col-9」はそれぞれ「col-9」「col-8」で指定します)

コントロール配置レイアウトのイメージ



楽々ウェブクライアントでデザインした結果は次のようになります。
テーブルと複数のサブフォーム間でレイアウトされていることを確認することができます。

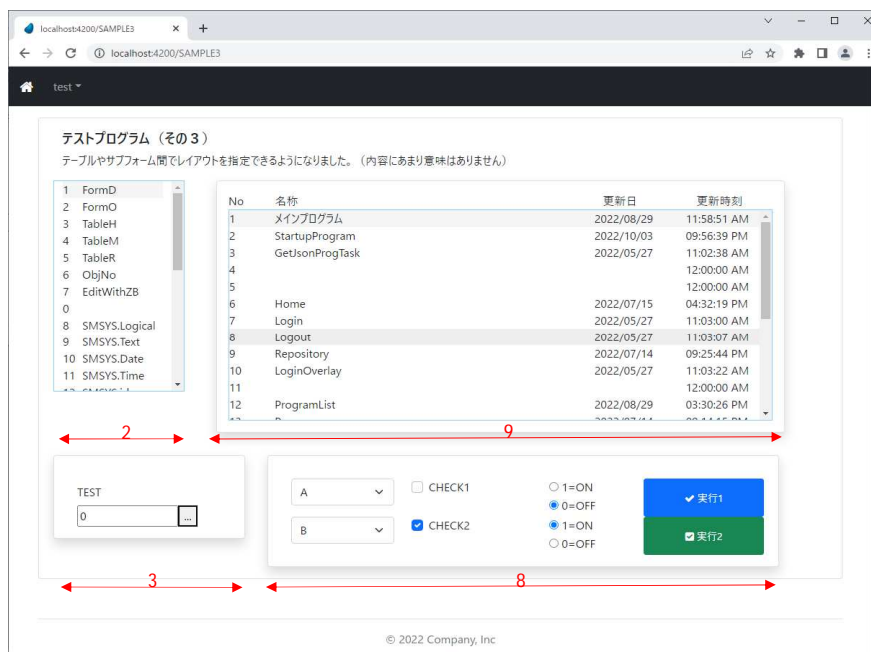
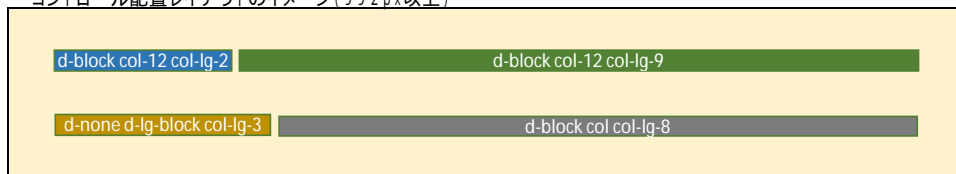


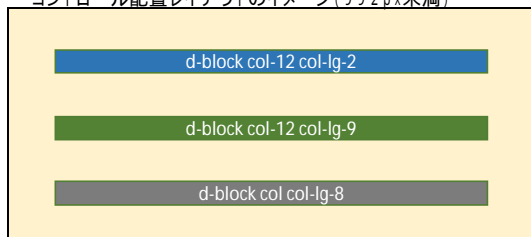
図6 楽々ウェブクライアント / デザイン例(テーブルと3つのサブフォームがあるプログラム)

bootstrapのグリッドシステムでは、画面サイズに応じてダイナミックに表示幅を指定することが可能です。
 例えば「PC等の画面サイズが広い場合は上記のようなレイアウトで構わないが、タブレットやスマートフォンの狭い環境ではコントロール別に縦に表示させたい」というようなケースです。
 また、PC用のサイトとスマートフォン用のサイトでは、表示させる広告が異なるかもしれません。
 従来のバージョンでは、閾値の選択は**Medium**(768px~)のみでしたが、**Small**(576px~)、**Large**(992px~)、**Extra large**(1200px~)の値も指定できるようになりました。
 また、これに加え、ブロック(コントロール)の並び順を変更する「order-N」の指定ができるようになっています。

コントロール配置レイアウトのイメージ(992px以上)



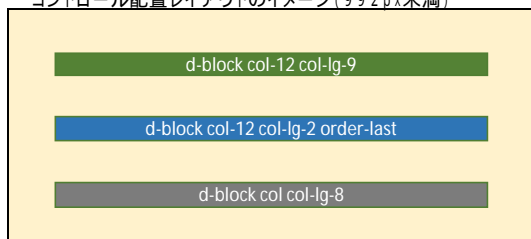
コントロール配置レイアウトのイメージ(992px未満)



先頭行(通常の並び)

2行目(先頭のカラムは非表示)

コントロール配置レイアウトのイメージ(992px未満)



先頭行(先頭カラムは「order-」

2行目(先頭のカラムは非表示)

楽々ウェブクライアントでデザインした結果は次のようになります。
 テーブルと複数のサブフォーム間でレイアウトされていることを確認することができます。

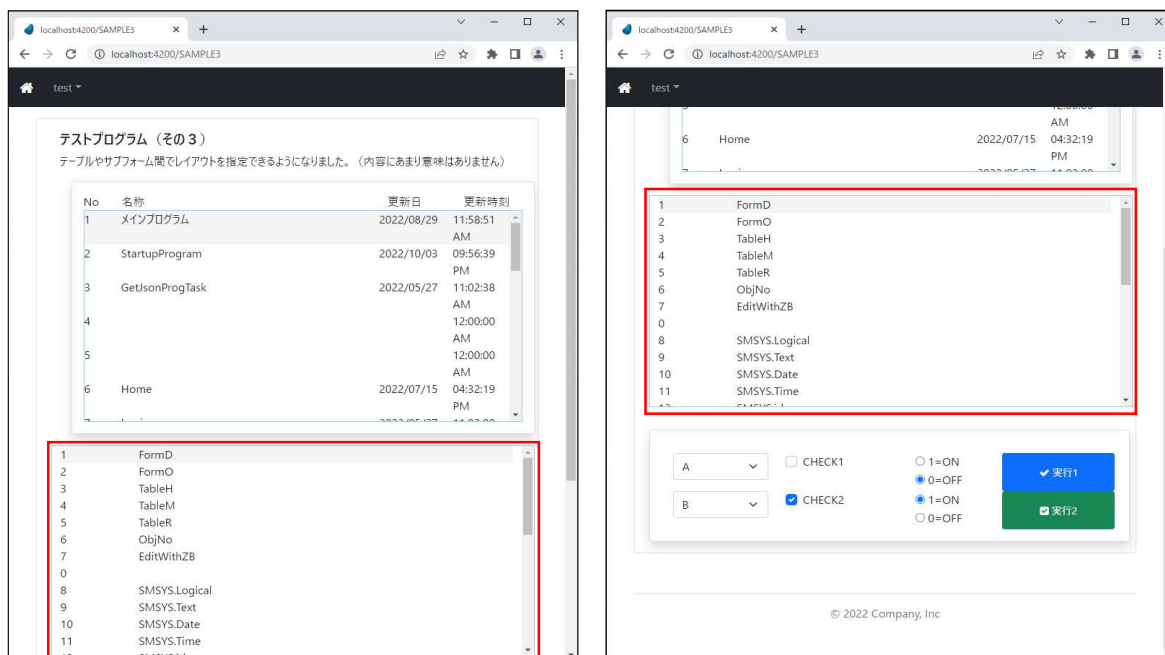


図7 楽々ウェブクライアント / デザイン例(画面サイズ992px未満で表示)

楽々ウェブクライアントの実際の設定方法について説明します。

イ) テーブルもしくはタブの場合

従来のバージョンでは、テーブルコントロールの属性を設定することができませんでしたが、今回のバージョンでは下記のような画面を表示して行うことができます。(タブの場合も同様の画面を表示します)

下図赤枠は、従来の仕様と変わった部分を補足しています。

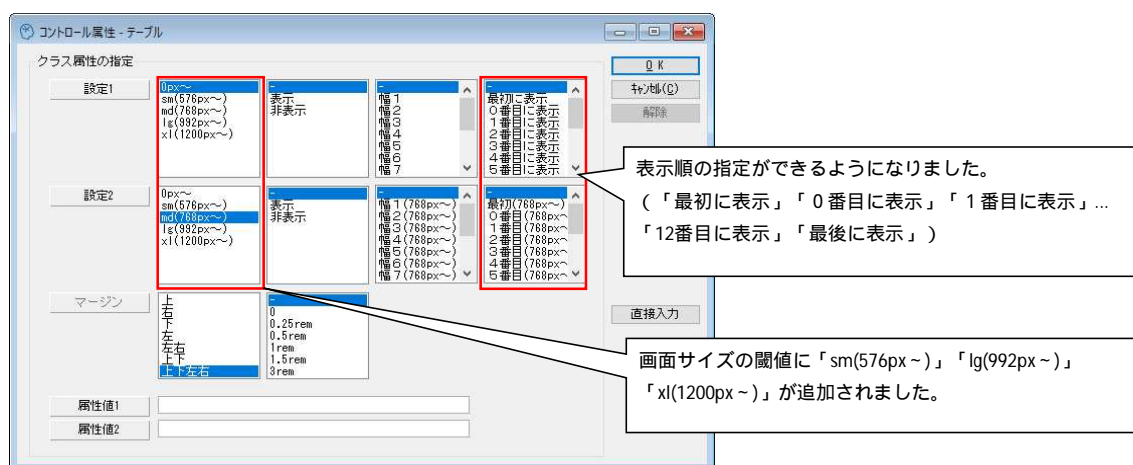


図8 テーブルコントロール属性設定画面

ロ) グループもしくはサブフォームの場合

グループやサブフォームの場合は、初期の画面では以前のバージョンと変わりありません。

が、「表示 / 非表示」ボタンを押したときの画面が拡張され、コラム幅や表示順の指定ができるようになりました。

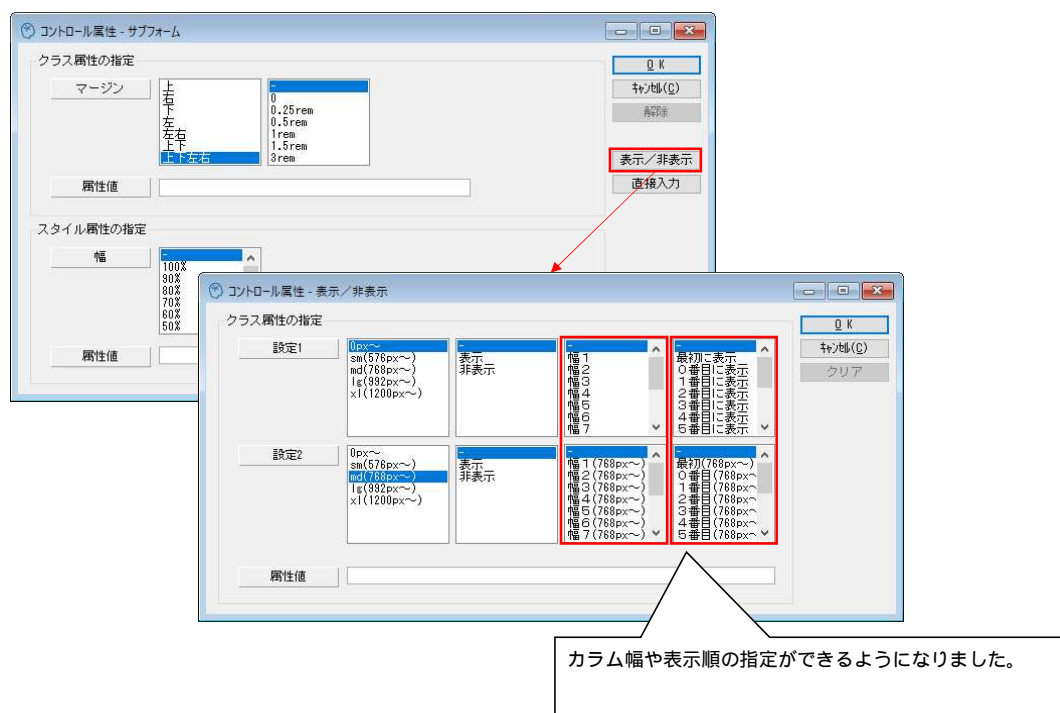


図9 サブフォームコントロールの楽々ウェブクライアント、属性設定画面

サンプルプログラム「GridTest」

新しい機能をテストするためのプログラムとして「GridTest」をサンプルプロジェクト内で提供しています。
設定値の詳細は、オンラインヘルプや同梱のPDFに記載しています。

オンラインヘルプ：プルダウンメニュー→「ヘルプ(H)」→「web上のヘルプ」
もしくは「<http://eternaldesign.jp/smsys/help/index.html>」
pdfドキュメント：楽々ウェブクライアント.pdf (2.5「サンプルプロジェクト」10「GridTest」)

イ) MAGICのフォーム

フォームは下図のように、テーブルとサブフォームを左右に並べています。
テーブルに表示されるリポジトリのタイプによりサブフォームに表示するプログラムを切り替えます。(クリックもしくはダブルクリックイベント)

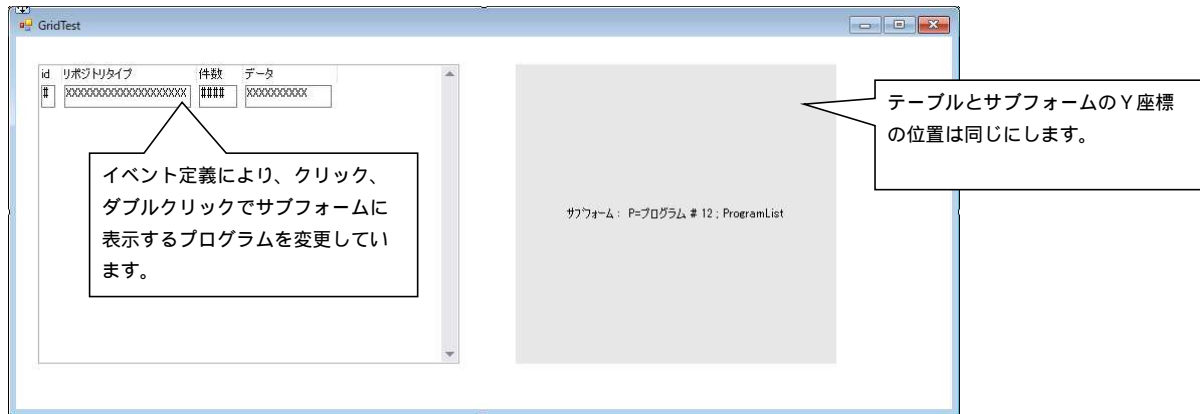


図10 サンプルプログラムのMAGICフォームエディタ画面

ロ) 楽々ウェブクライアントの設定(コントロール属性)

楽々ウェブクライアントの設定例を見てみましょう。

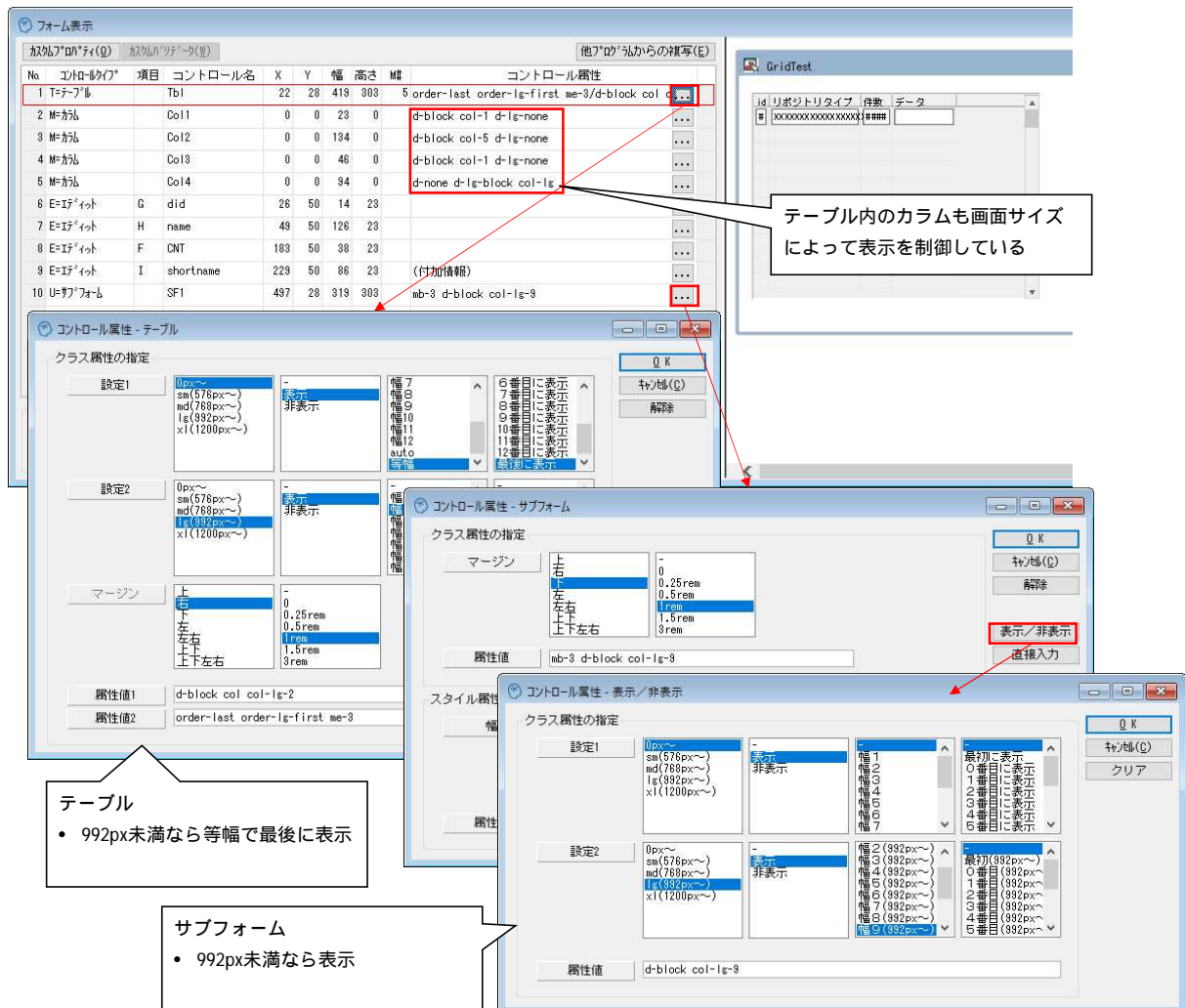


図11 サンプルプログラム / コントロール属性の設定

ハ) 楽々ウェブクライアントの設定(テンプレートの変換)

「変換ルール」は新しく追加されたルール「サンプル CARD変換(グリッド利用)」を選択します。

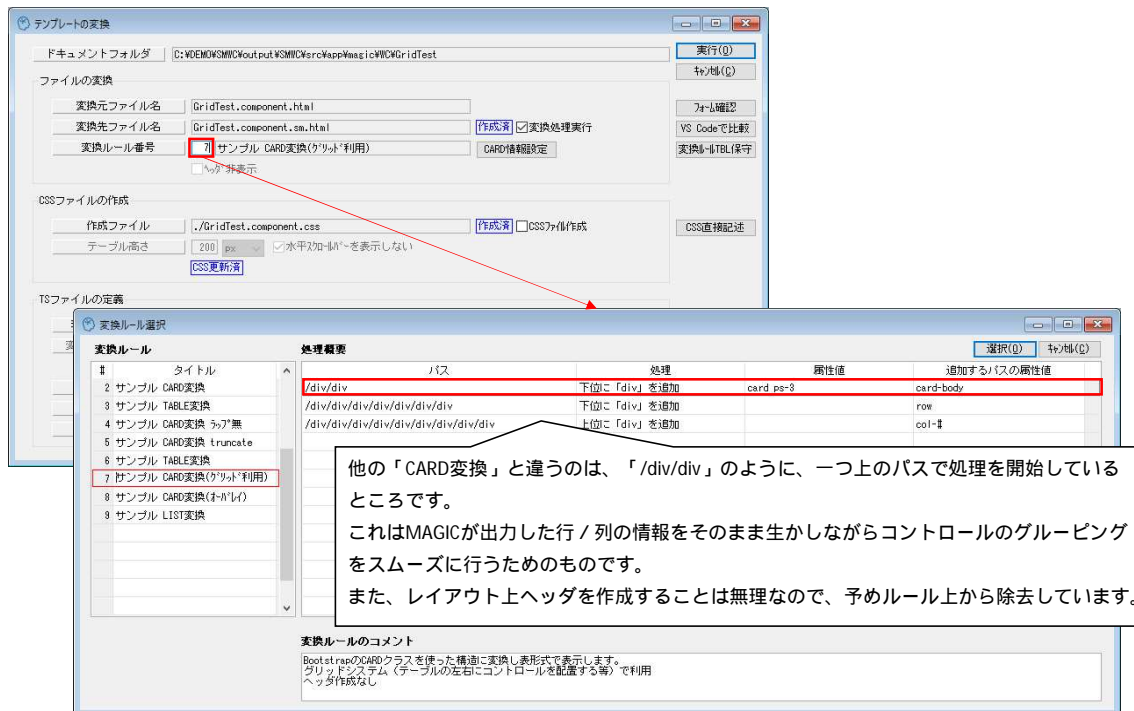


図12 サンプルプログラム / テンプレートの変換 / 変換ルール選択

変換結果は下図に示します。

クラス「row」のdiv要素の配下に、テーブルとグループが配置されていることが分かります。(外側の赤枠)

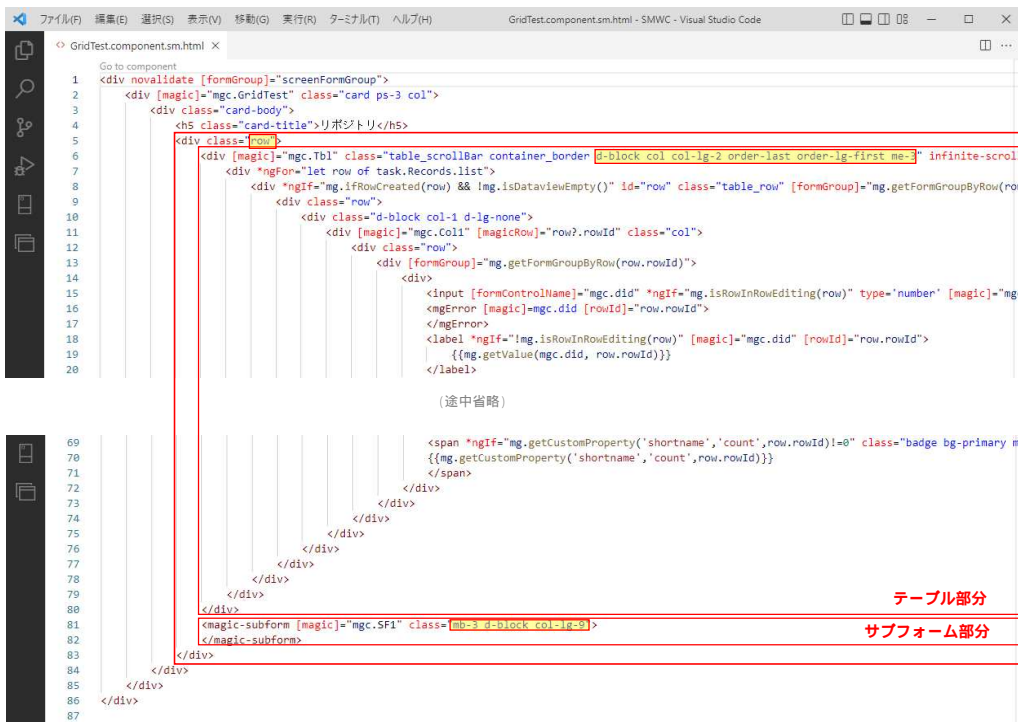


図13 Visual Studio Codeによる結果確認(GridTest.component.sm.html)

< 補足 >

今回のバージョンでは、MAGICが出力した方向を示すスタイル属性を自動的にクラス属性に変換するように、仕様が変わりました。

例) style="display: flex; flex-direction: column" → class="col"
style="display: flex; flex-direction: row" → class="row"

この仕様変更により、変換ルールの定義から「style」属性を除去する指定が削除されました。

二) 実行結果

画面サイズ 992px以上のときと、992px未満のときの実行結果をそれぞれ示します。

リポジトリ

プログラム

位置付 0 ☐ フォルダ選択

番号	名前	フォルダ	公開名	最終更新日	時刻
1	メインプログラム			2022/08/29	11:58 AM
2	StartupProgram			2022/09/30	10:51 AM
3	GetJsonProgTask	BATCH		2022/05/27	11:02 AM
4		BATCH			
5		BATCH			
6	Home	WC		2022/07/15	04:32 PM
7	Login	WC		2022/05/27	11:03 AM
8	Logout	WC		2022/05/27	11:03 AM
9	Repository	WC		2022/07/14	09:25 PM
10	LoginOverlay	WC		2022/05/27	11:03 AM
11		WC			
12	ProgramList	WC		2022/08/29	03:30 PM

col-lg-2 col-lg-9

© 2022 EternalDesign Corp.

テーブル（赤枠）とサブフォーム（青枠）が、左右に 2 : 9 の比率で表示

図14 実行結果 (画面サイズ 992px以上)

リポジトリ

プログラム

位置付 0 ☐ フォルダ選択

番号	名前	フォルダ	公開名	最終更新日	時刻
1	メインプログラム			2022/08/29	11:58 AM
2	StartupProgram			2022/09/30	10:51 AM
3	GetJsonProgTask	BATCH		2022/05/27	11:02 AM
4		BATCH			
5		BATCH			
6	Home	WC		2022/07/15	04:32 PM
7	Login	WC		2022/05/27	11:03 AM
8	Logout	WC		2022/05/27	11:03 AM
9	Repository	WC		2022/07/14	09:25 PM
10	LoginOverlay	WC		2022/05/27	11:03 AM
11		WC			
12	ProgramList	WC		2022/08/29	03:30 PM

1	モデル	7
2	データ	2
3	プログラム	21
4	ヘルプ	0
5	権利	0
6	メニュー	1
7	コンポーネント	2

order-last の指定により最後に表示

テーブル（赤枠）とサブフォーム（青枠）は全幅で表示

表示順の指定があるため、テーブル（赤枠）は後ろに表示

図14 実行結果 (画面サイズ 992px未満)

2. その他の改善事項

フォームイメージ表示用フォントの指定ができるようになりました。

WebClientタスクでは、1番のフォントを使用して描画を行っていましたが、環境設定プログラムでフォント番号を指定できるようになりました。

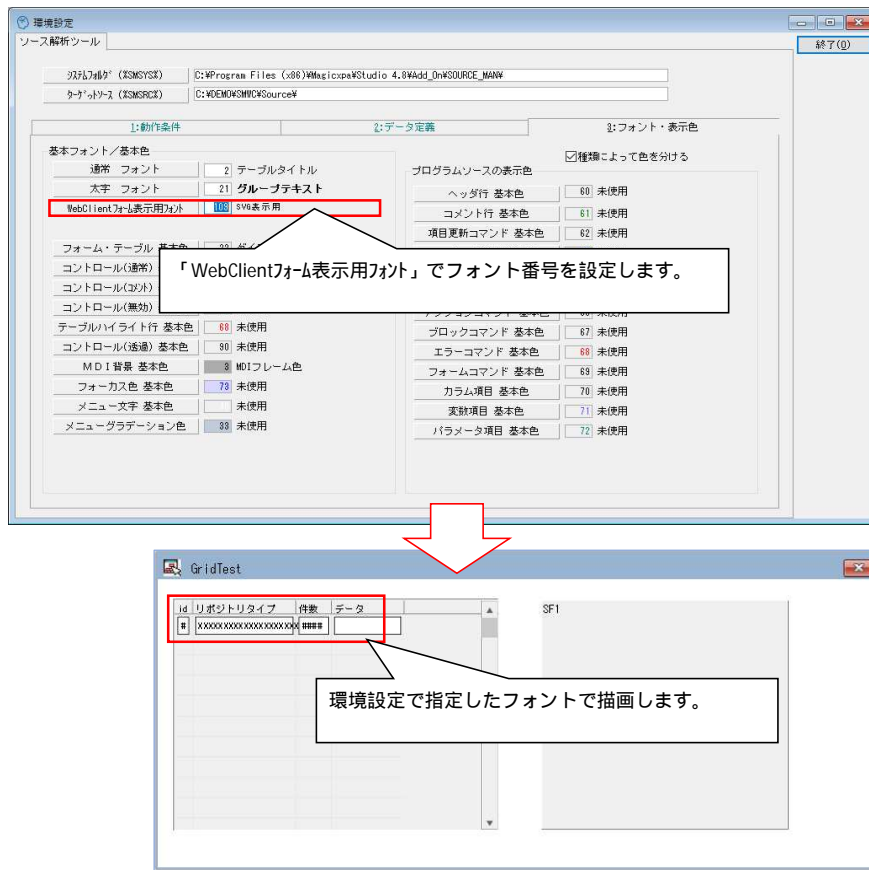


図15 環境設定画面とフォームのイメージ表示

フォーム表示画面の改善

コントロール属性呼出し用のボタンにパーキングするようし、キー入力により連続して呼び出せるようになりました。

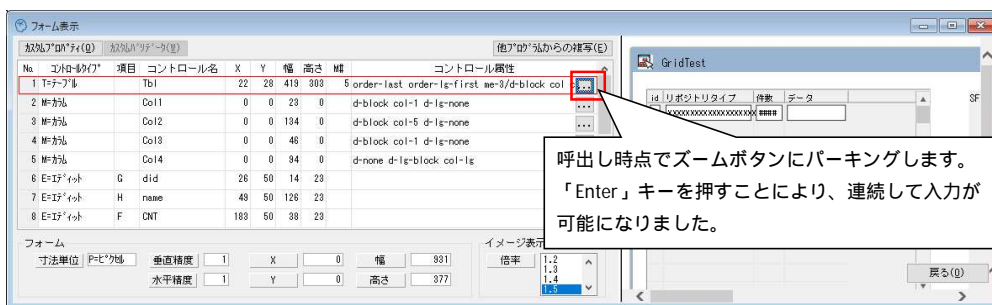


図16 フォーム表示画面のコントロール属性編集ボタン

テンプレートの変換の改善

選択した変換ルールへのヘッダ定義の有無を取得し、ヘッダが無い場合は「ヘッダ非表示」チェックボックスを無効に、またチェック済なら外すように修正しました。

定数定義ファイル (ConstFile.xml) のチェック処理

定数ファイルのバージョンを起動時にチェックし、起動中のシステムの要求するバージョンとミスマッチしている場合は起動時に警告メッセージを表示するようにしました。

表示メッセージ: ConstFile.xml のバージョンが一致していません。

もし、上記のメッセージが出たときは、システムのアップデートに失敗しています。原因としては、キャビネットファイル (*.ECF) のみ更新した場合等が考えられますので、再度、セットアップ用プロジェクトによりシステムを更新して下さい。

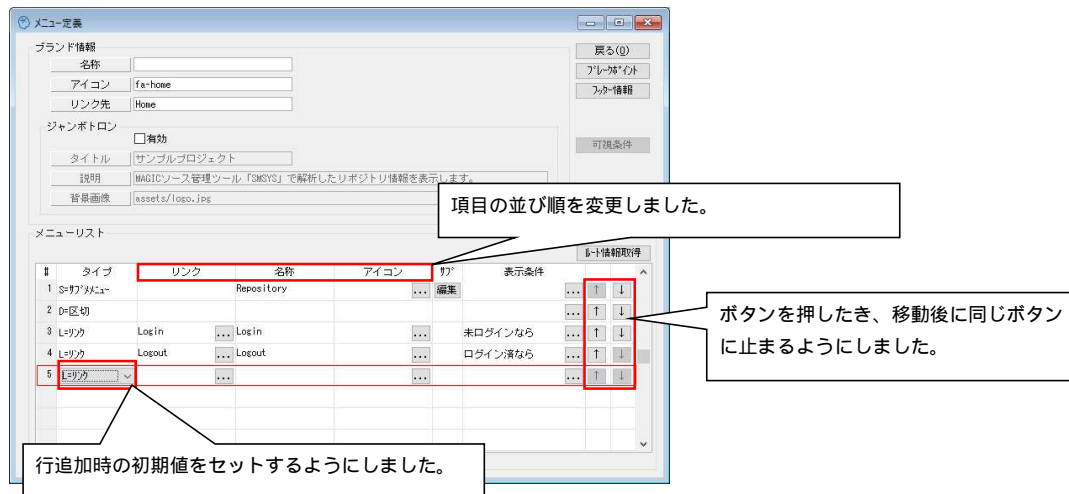
ナビゲーションバー / メニュー定義明細編集の改善

メニューリストで行を追加したときに、デフォルト値として「L=リンク」をセットするようにしました。

項目の並びを「名称 アイコン リンク」から「リンク 名称 アイコン」に変更しました。これは、「リンク」でズームしルートの一覧画面から選択した時に、選択したルート名を名称にセットしていますが(従来からの機能)、この流れで入力をしやすくするためです。

ズームできる項目にはズームボタンを付加しました。

並び順変更用のボタン「↑」「↓」を押したときは、並び順の表示を更新した後、押したボタンにカーソルを飛ばすようにしました。これにより、連続して同じ行を移動させることができるようにしました。



楽々ウェブクライアント / SMSYS Ver 1.03新機能の紹介

1. テーブルに配置したエディットコントロールの右寄せ機能

テーブルに配置した数値項目等のエディットコントロールを右寄せ指定できるようになりました。但し、ここでの右寄せ指定可能なエディットコントロールとは、「テーブルに配置した編集状態にないエディット項目」です。また、エディットコントロールの一覧を確認し、一括で右寄せ指定する機能を追加しています。

エディットコントロールの属性設定ダイアログ

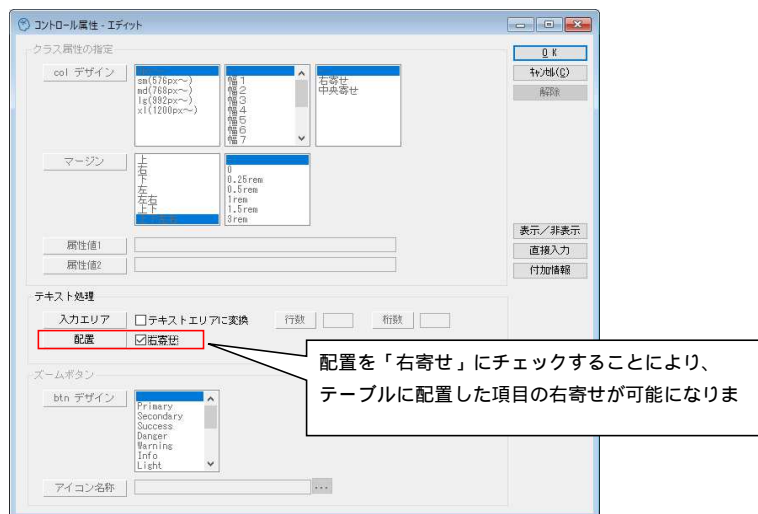


図1 改良されたエディットコントロールのコントロール属性設定画面

フォーム表示プログラムに「数値項目右寄せ」ボタンを追加

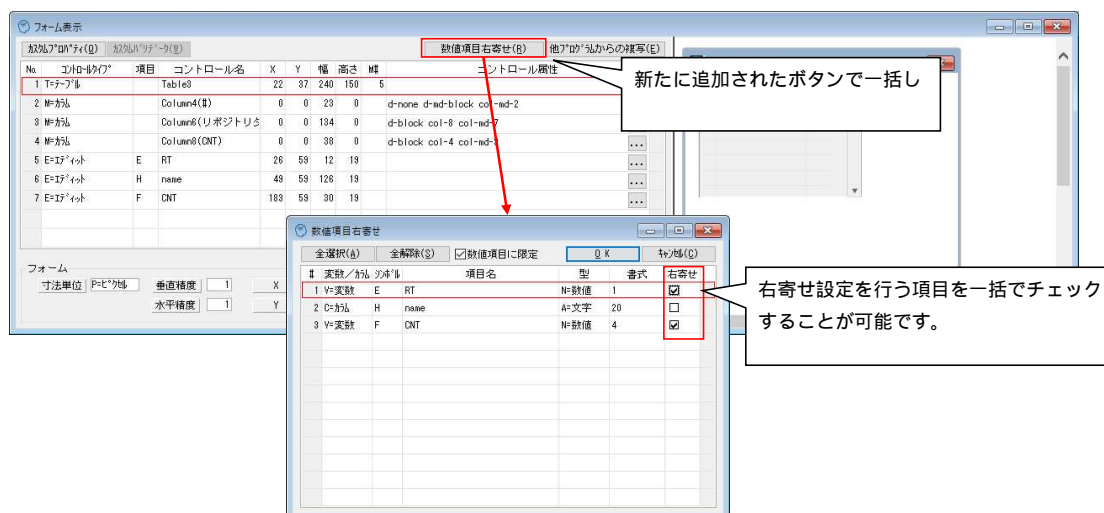


図2 追加された「数値項目右寄せ」ボタンとその設定画面

実行結果

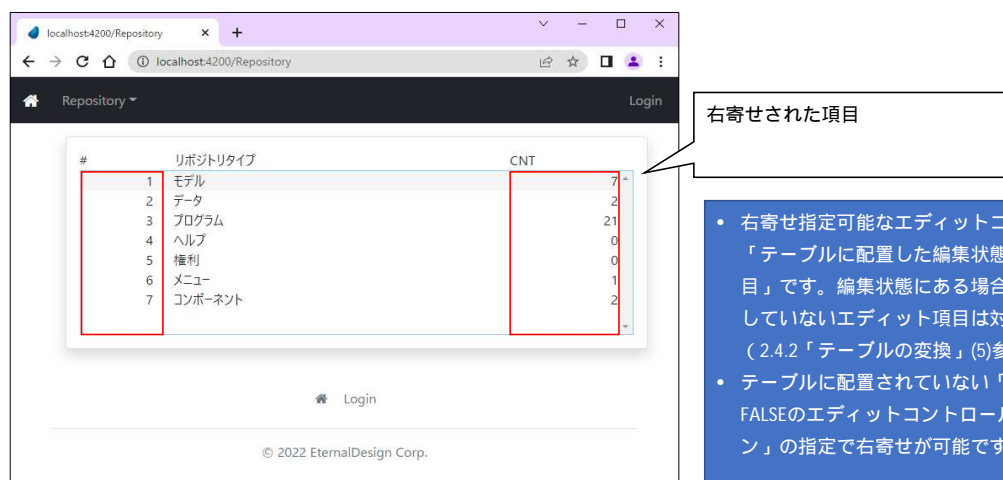


図3 右寄せ指定したときの実行画面

- 右寄せ指定可能なエディットコントロールは、「テーブルに配置した編集状態にないエディット項目」です。編集状態にある場合や、テーブルに配置していないエディット項目は対象外です（2.4.2「テーブルの変換」(5)参照）
- テーブルに配置されていない「修正許可」属性がFALSEのエディットコントロールは、「Colデザイン」の指定で右寄せが可能です

ソースを確認

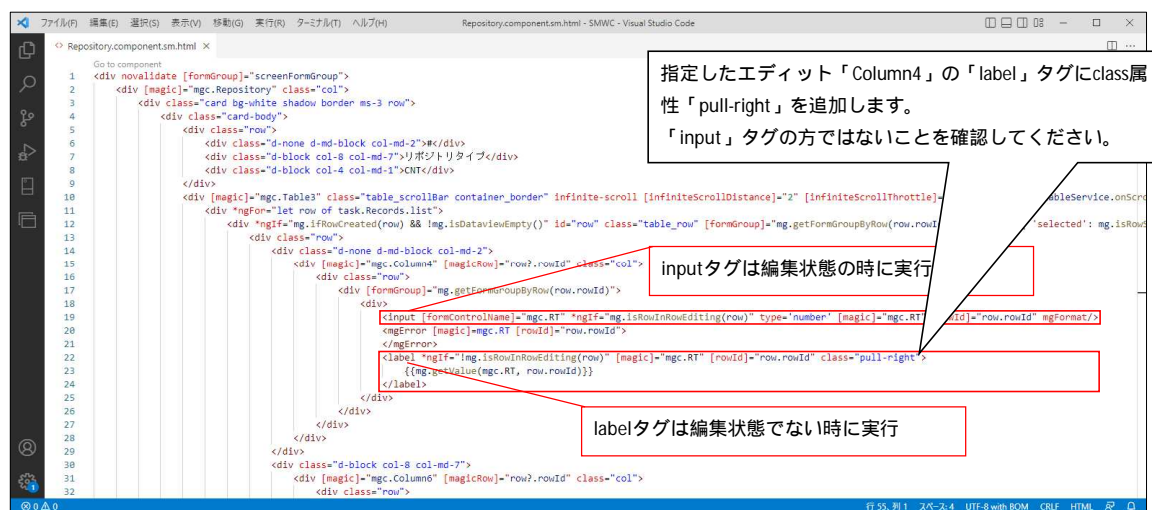


図4 右寄せ指定を行ったときのHTML

2. テーブルカラムのコントロール名表示にタイトルと一緒に表示するように修正(「フォーム表示」画面)

「フォーム表示」画面において、カラムコントロールのコントロール名の表示は、列が多いテーブルの場合に判別が難しかったので、表示する情報を変更しました。

従来 カラムコントロールのコントロール名を表示
新仕様 「コントロール名(カラムテキスト)」の書式で両者を表示

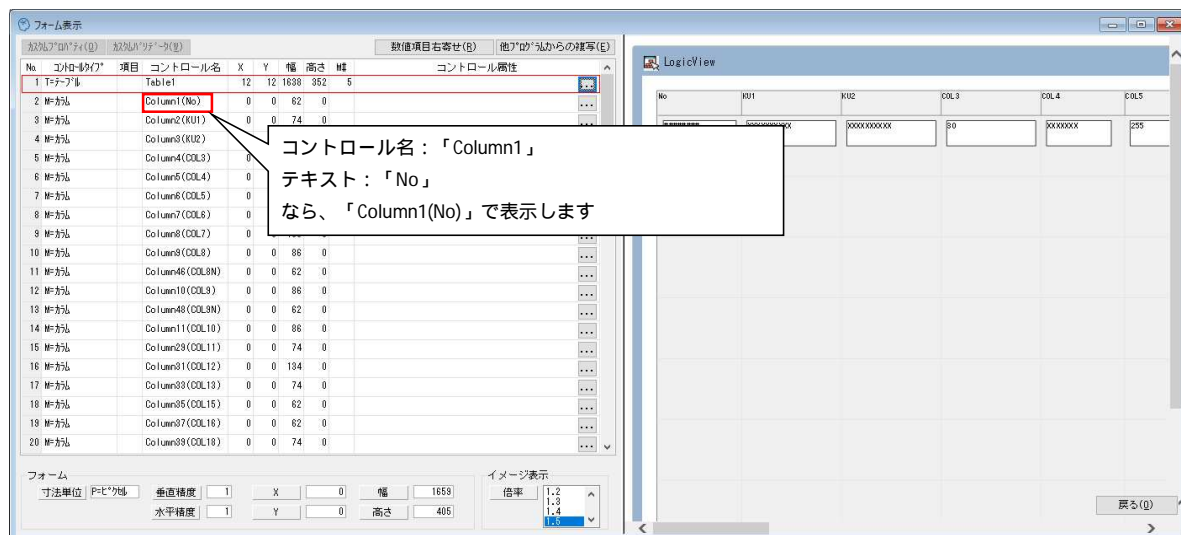


図5 フォーム表示画面におけるカラムコントロールの「コントロール名」表示

3. グループ、サブフォーム、タブ、テーブルに対する付加情報入力

グループ、サブフォーム、タブ、テーブルに対しても付加情報を入力できるようにしました。但し、これらのコントロールの場合は「B=前置き」は選択可ですが、「A=後置き」は選択不能としました。

追加された「付加情報」ボタン

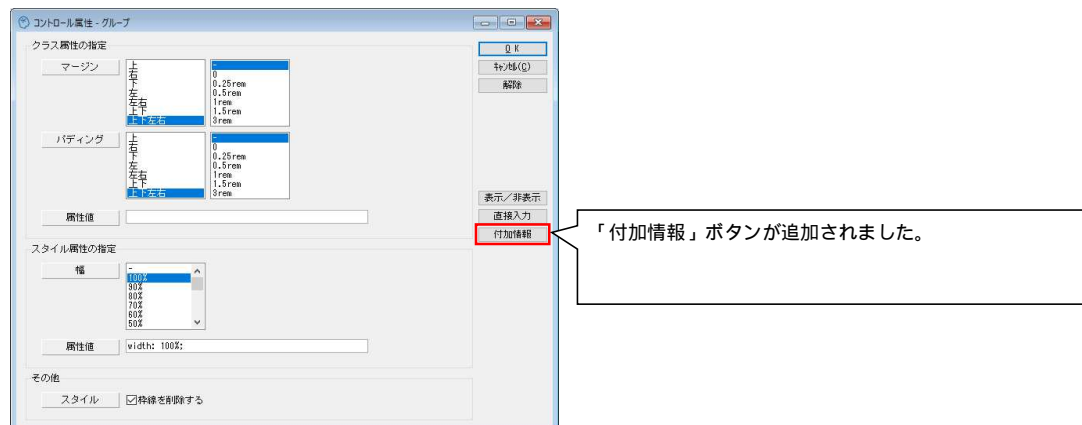


図6 追加された「付加情報」ボタン

入力例

プログラム「home」のグループコントロールを選択し、「付加情報」ボタンを押します。
このプログラムはカスタムプロパティが設定してあるので、入力アシスト付きのフォームが表示されます。

カスタムプロパティが設定されている場合は、「B=構文追加（前置）」以外に、「D=要素の表示条件」等の選択ができません。

```
<div class="row" *ngIf="mg.getCustomProperty('Home','user')!=0">
  <label class="label_overflow col-md-3 col-form-label text-md-end me-2">
    USER
  </label>
  <label class="col-md-3 col-form-label">
    {{mg.getCustomProperty('Home','user')}}
  </label>
</div>
```

図7 付加情報の入力例(カスタムプロパティの入力アシスト付きフォーム)

実行結果

ログイン後、ユーザIDを表示します。

図8 付加情報を入力した時の実行結果

ソースを確認

挿入された構文
(グループコントロール記載部分
の前に配置)

グループコントロール記載部分

図9 グループコントロールに「前置き」で付加情報を入力したHTML

4. その他の修正事項

フォームのイメージ作成時の描画を改善しました。

フォーム表示では、フォームエディタ上の画面をSVGを使って表示していますが、その表示具合を若干修正しました。

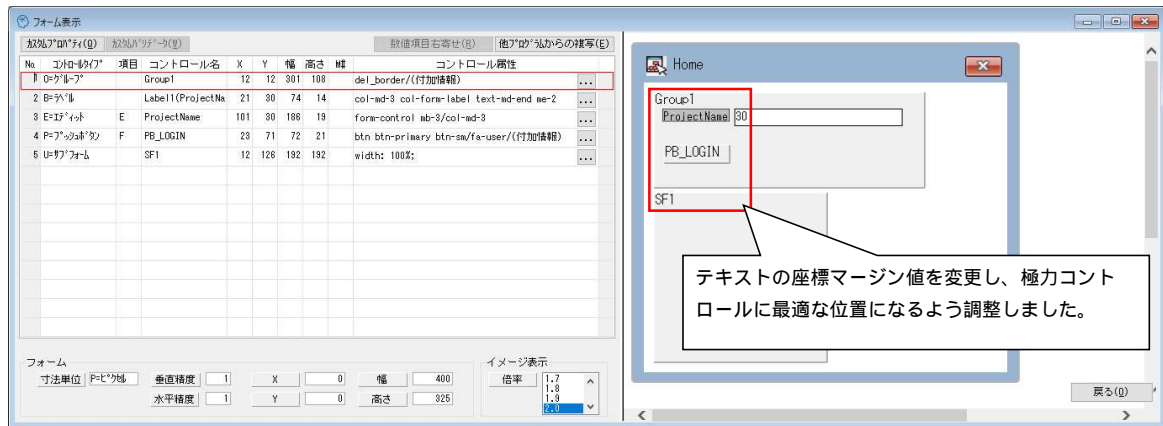


図10 SVGによるフォームイメージ表示

ラベルのcolデザインで整列方法に中央寄せを追加しました。

ラベルの整列は「右寄せ」でしたが、これに「中央寄せ」を追加しました。



図11 ラベルコントロールの整列

エディットコントロールも「修正許可」プロパティがFALSEのときはラベル扱いとなります。
このとき、ラベルと同じように「col デザイン」で「中央寄せ」の指定が可能です。

「テンプレートの変換」実行終了時のダイアログ

いままでは実行ボタンを押したとき、終了時に「処理終了！」というダイアログを出していましたが、これをやめて直ちに終了するようにしました。
代わりにWindowsの通知欄に下記のようなメッセージを出力します。

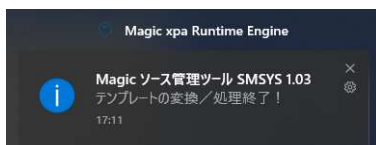


図12 Windowsの通知による処理終了メッセージ

楽々ウェブクライアント / SMSYS Ver 1.04新機能の紹介

1. ナビゲーションバーを作成しない利用を可能に

今までのバージョンでは、Bootstrap設定ウィザードにより、予め設定したテンプレートを利用することによって、ナビゲーションバーを簡単に作成することが出来ました。

しかし、自前でスタートアッププログラムを作成したいような場合は、もしかしたら余計な機能になるかもしれません。

あるいは、当アプリケーションのフォーム変換機能を使用し、スタートアッププログラムを作成したいと思うようなケースもあるかと思います。

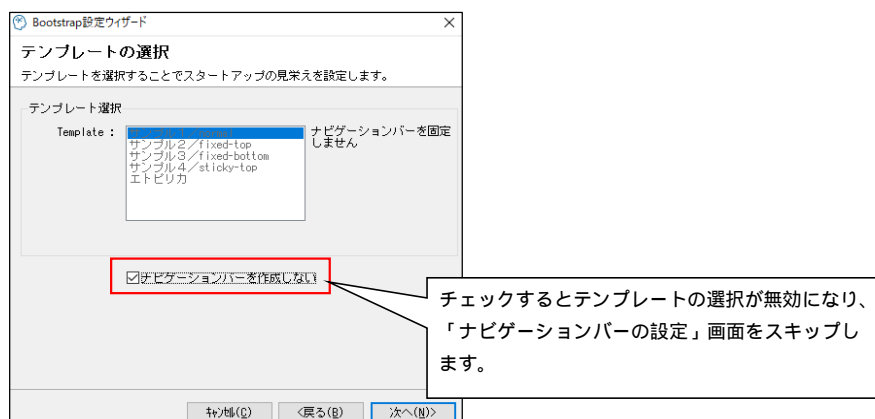
今回のバージョンでは、このテンプレートによるウィザード機能をスキップするオプションを設置することにより、ナビゲーションバーを作成する場合は従来通り、作成し場合は、テンプレートの変換を呼ぶこともでき、全く何もしないことも選択できる・・・というように、その自由度を広げました。

ナビゲーションバーの作成	変換時、実行するプログラム	処理対象のフォームのデザイン	ウィザード時の処理
作成する	Navbar作成専用処理	デフォルトアウトレットの名称のみ参照	あり
作成しない	「テンプレートの変換」プログラム	フォームに配置した全てのコントロールを処理	なし

< 変更箇所 >

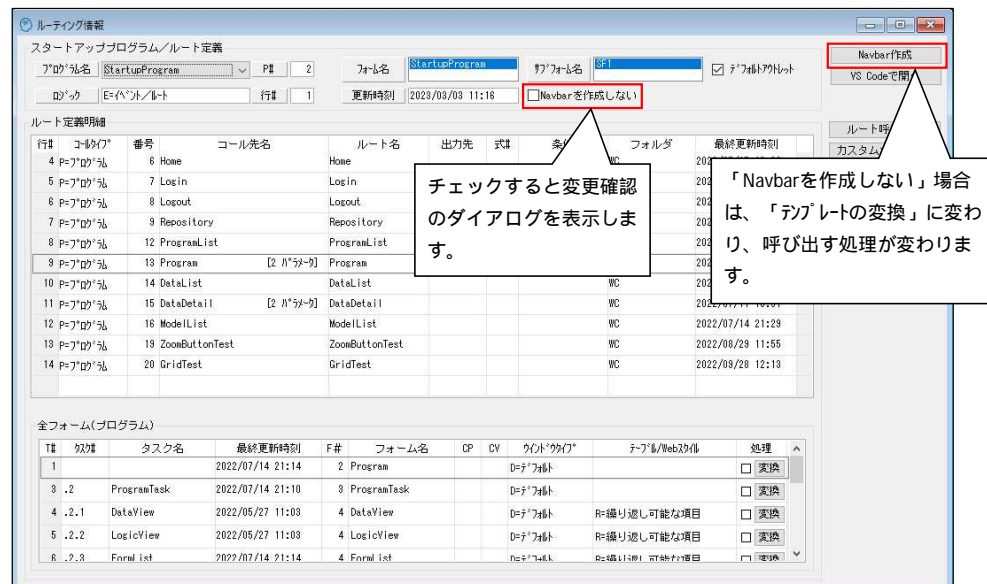
Bootstrap設定ウィザードの「テンプレート選択」画面に「ナビゲーションバーを作成しない」オプションを追加しました。

設定時に指定する場合はウィザードのチェックボックスをチェックにします



ルーティング情報画面に「Navbarを作成しない」チェックボックスを設置しました。

後で変更する場合は「ルーティング情報」プログラムを呼び出して行うことが可能です。

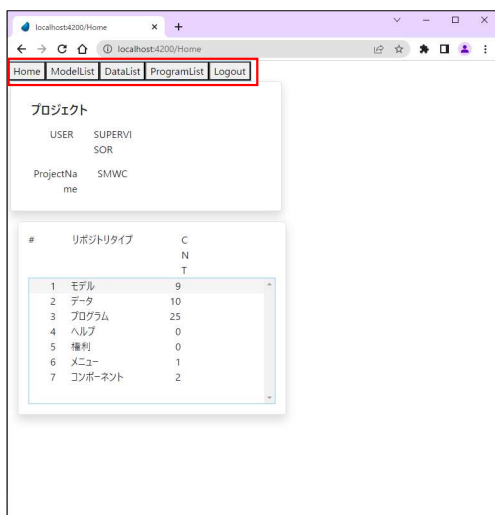


WebClientフォーム画面からも呼び出すプログラムが切り替わります。

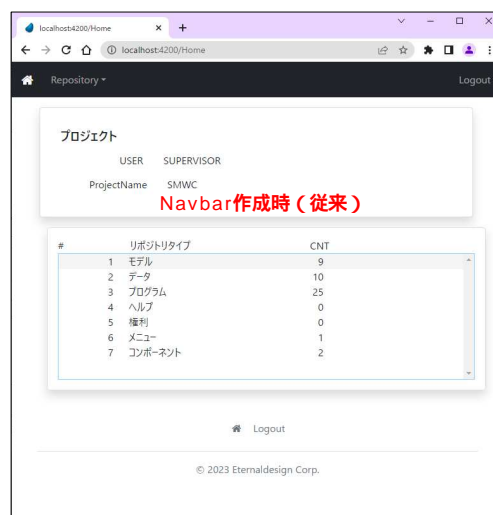
WebClientフォーム											
VS Codeで開く											
#	プログラム名	モード	フォルダ	処理	タスク名	最終更新時刻	F#	フォーム名	CP	DV	処理
1	StartupProgram					2023/05/12 17:21	2	StartupProgram		D=デフォルト	変換
6	Home	<input checked="" type="checkbox"/>	WC			2022/07/15 16:32	2	Home	...	D=デフォルト	<input checked="" type="checkbox"/> 変換
7	Login	<input checked="" type="checkbox"/>	WC			2022/05/27 11:03	2	Login		D=デフォルト	<input checked="" type="checkbox"/> 変換
8	Logout	<input checked="" type="checkbox"/>	WC			2022/05/27 11:03	2	Logout		D=デフォルト	<input type="checkbox"/> 変換
9	Repository	<input checked="" type="checkbox"/>	WC			2022/07/14 21:25	2	Repository			<input checked="" type="checkbox"/> 変換
10	LoginOverlay	<input type="checkbox"/>	WC			2022/05/27 11:03	2	LoginOverlay			<input checked="" type="checkbox"/> 変換
12	ProgramList	<input checked="" type="checkbox"/>	WC			2022/06/29 15:30	2	ProgramList			<input checked="" type="checkbox"/> 変換
12	ProgramTbl		WC	.1	ProgramTbl	2022/06/29 15:30	3	ProgramTbl			<input checked="" type="checkbox"/> 変換
13	Program	<input checked="" type="checkbox"/>	WC			2022/07/14 21:14	2	Program			<input checked="" type="checkbox"/> 変換
13	ProgramTask		WC	.2	ProgramTask	2022/07/14 21:10	3	ProgramTask			<input checked="" type="checkbox"/> 変換
13	DataView		WC	.2.1	DataView	2022/05/27 11:03	4	DataView			<input checked="" type="checkbox"/> 変換
13	LogicView		WC	.2.2	LogicView	2022/05/27 11:03	4	LogicView			<input checked="" type="checkbox"/> 変換
13	FormList		WC	.2.3	FormList	2022/07/14 21:14	4	FormList			<input checked="" type="checkbox"/> 変換
14	DataList	<input checked="" type="checkbox"/>	WC			2022/07/14 21:15	2	DataList			<input checked="" type="checkbox"/> 変換
14	DataTbl		WC	.1	DataTbl	2022/07/11 16:43	3	DataTbl		R=繰り返し可能な項目	<input checked="" type="checkbox"/> 変換
15	DataDetail	<input checked="" type="checkbox"/>	WC			2022/07/11 16:31	2	DataDetail		D=デフォルト	<input checked="" type="checkbox"/> 変換
15	DataCol		WC	.1	DataCol	2022/07/11 16:20	3	DataCol		R=繰り返し可能な項目	<input checked="" type="checkbox"/> 変換
15	DataIndex		WC	.2	DataIndex	2022/07/11 16:15	3	DataIndex		D=デフォルト	<input checked="" type="checkbox"/> 変換
16	ModelList	<input checked="" type="checkbox"/>	WC			2022/07/14 21:29	2	ModelList		D=デフォルト	<input checked="" type="checkbox"/> 変換

スタートアッププログラムの場合は、どちらのモードになっているかにより、呼び出される処理が変わります。

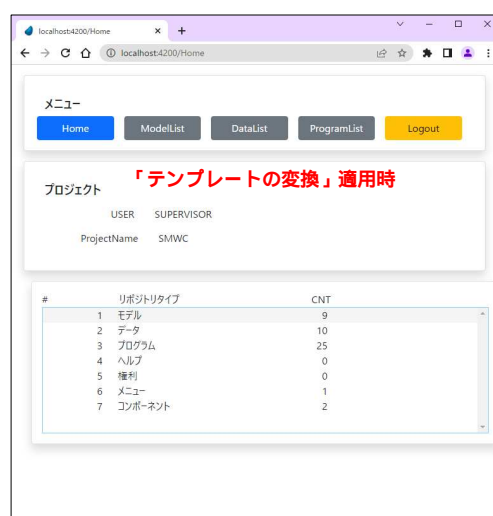
<実行例>



「ナビゲーションバーを作成」時
(初期値)



「ナビゲーションバーを作成しない」選択時



2. その他の修正事項

< 楽々ウェブクライアント >

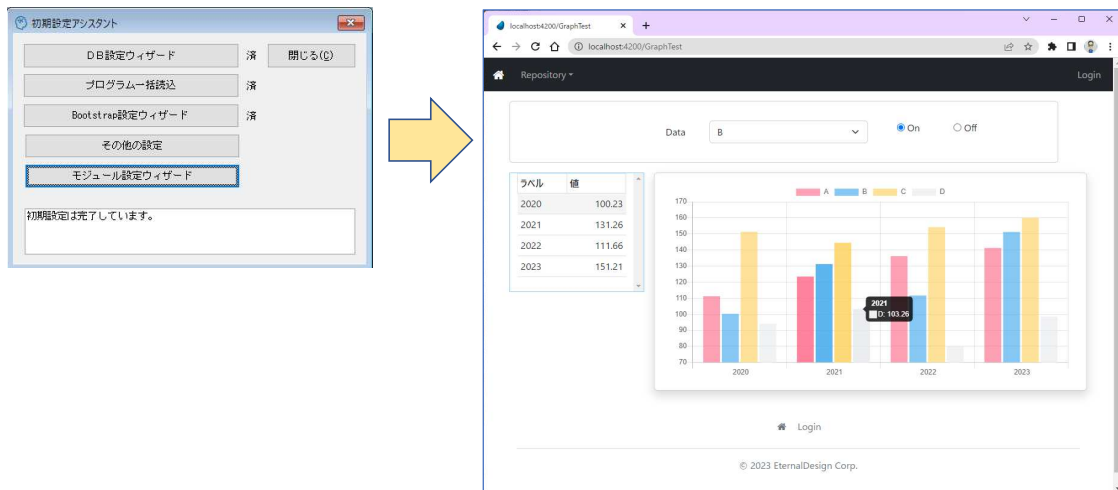
変換ルールの編集結果保存の仕様を変更

変換ルールの保存ファイルの仕様を変更し、システムで提供する変換ルールとユーザ側で編集する変換ルールを分離し、バージョンの更新があっても個々に内容を保持できるようにしました。



モジュール設定ウィザード

初期設定アシスタントに「モジュール設定ウィザード」を追加しました。提供する定義ファイルを組み込むことにより、Angularモジュールの組み込み等をウィザードで行います。
(定義ファイルは順次提供していく予定です)



一括更新の不具合修正

WebClientフォームの「一括更新」を使用し、復元処理を行ったときに、テーブルの高さ指定等のCSS設定内容が反映されない不具合を修正しました。

< SMSYS >

SQLite指定時の不具合修正

データベースの「DB設定」で「SQLite」を選択したときに、プログラム解析時点でエラーが出て処理が中断してしまう不具合を修正しました。

楽々ウェブクライアント のお問い合わせ

楽々ウェブクライアントの機能、ライセンス体系、価格、お見積り依頼、テクニカル サポートなど、本製品に関するご質問やご不明な点がありましたら、お気軽にお問い合わせください。

お問い合わせ方法 下記の項目について記載し、メールでご送信下さい。

お問い合わせ内容の例	製品価格 / ライセンス 見積作成依頼 購入に関するお問い合わせ テクニカルサポート その他
ご記載頂くお客様情報	会社名 部署名 氏名 Email アドレス
送付先	support@eternaldesign.jp