Part 1

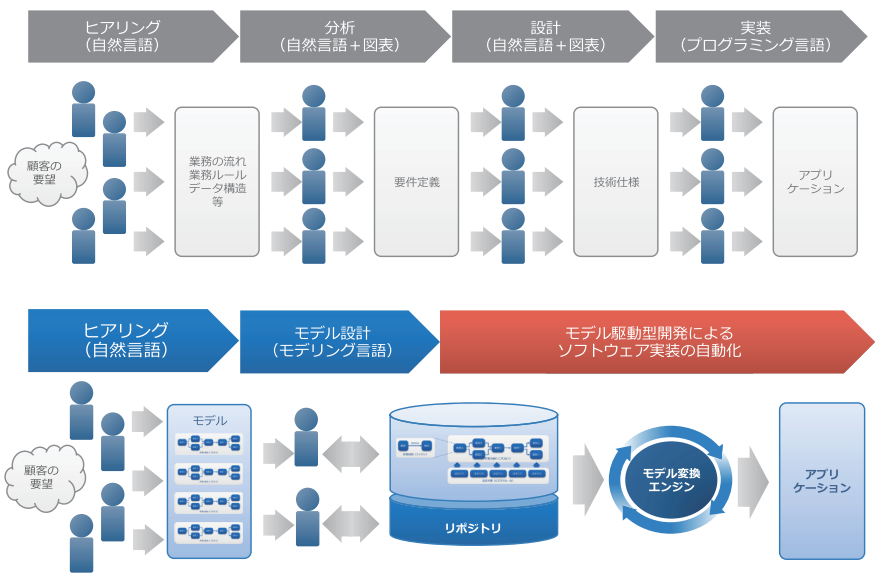
CHAPTER 2

モデル駆動型開発とは

モデル駆動型開発とは

従来のスクラッチ開発とモデル駆動型開発の違いはなに？

モデル駆動型開発の最大の特徴は、モデルと呼ばれる設計情報をモデリング言語で記述することで、ソフトウェアを自動で製造できることです。従来の開発では、自然言語と図表を用いて設計を行っていたため、プログラマの経験と勘で、これらの設計情報をプログラミング言語に変換していました。



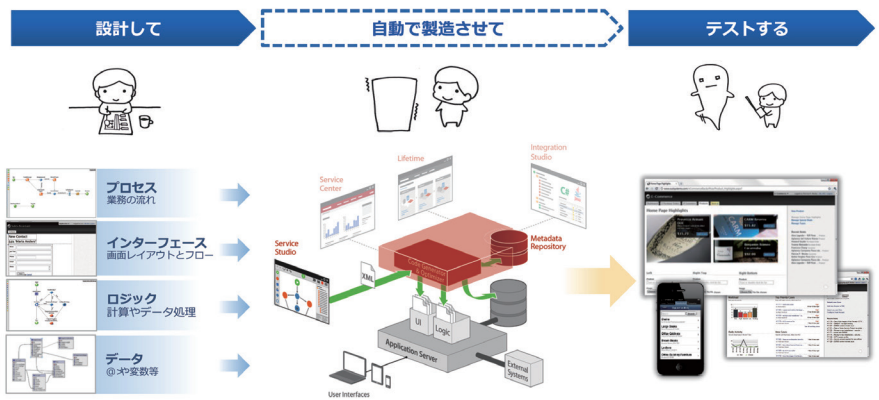
モデリング言語ってなに？

モデリング言語とは、あらかじめ決められた文字列や記号を用いて、情報や知識、機能等の要素を表現できる言語です。システム開発には多くのモデリング言語がありますが、もっとも有名なのは、UML（Universal Modeling Language）です。UML 2.0 では13種類の図を用いて、様々なシステムを表現することができ、ソフトウェア開発においては、汎用モデリング言語として広く利用されています。最近ではUMLを用いてモデル駆動型開発を行う研究もありますが、モデルの記述方法の自由度が高いため、そのモデルから完全に動作するソフトウェアを製造することは難しいと考えらえています。

本演習では、業務アプリケーションのソフトウェア開発に特化したOutSystems社のビジュアルモデリング言語を用いることで完全に動作するソフトウェアの自動生成を行っています。

OutSystems Platformとは？

OutSystems Platform とは、第五世代言語であるビジュアルモデリング言語を用いて、モデル駆動型開発を行うことができるソフトウェア開発基盤です。OutSystems Platformでは、プロセスとインターフェース、ロジック、データの４種類のモデルを設計することにより、ソフトウェアの製造工程を自動化します。また設計されたモデルを活用することで、保守及び運用機能の自動化も実現しています。



OutSystems Platformはプロフェッショナル向けのツール

OutSystems Platformは、システムの開発から保守、運用までをカバーしており、Webアプリケーションを開発するための基本機能を全て保有しています。このため、すべての機能を使いこなすためには、データベースをはじめとした様々な技術の知識が必要となります。

C#とビジュアルモデリングの比較

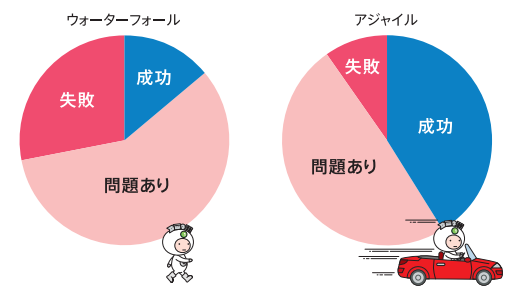
同じ処理を行うロジックをC#とビジュアルモデリングで表現すると下記のようになります。数十行のコードの塊をアイコンで表現するようなイメージとなるため、プログラミングを経験した方であれば、直感的にモデリングを行うことが可能です。

アジャイル開発ってなに？

モデル駆動型開発を行う場合に最も適したプロジェクト管理の手法は、アジャイルだと考えられています。従来のウォーターフォール開発では、プロジェクトマネージャによって開発がトップダウンで管理されているのに対し、アジャイル開発ではチームが自律的に活動しながら開発を進めていくというスタイルが大きく異なります。またアジャイルではリリースを細かく細分化し、リリース単位で様々な仕様変更を積極的に取り入れることで、より良いソフトウェアを開発できるようなプロセスになっています。ウォータフォールは、システム開発において最もリスクの大きかったやり直しや手戻りが発生しないように考えらえた管理手法だったのですが、最近の統計ではアジャイルを採用したプロジェクトの成功率がウォーターフォールの3倍という結果になっています。

ウォーターフォール	アジャイル
成果物と計画に着目	顧客の価値とフローに着目
プロジェクトマネージャによって管理される	チームは自立的に活動する
仕様変更は厳格な変更管理プロセスによって拒否される	仕様変更は期待され歓迎される
顧客は月次の定例や承認で限定的に参加する	顧客は業務変更やソフトウェア開発に積極的に参加する
事前に詳細な計画を立てる	前進しながら計画を立てていく
マネージャが交渉してスコープ主体でデリバリを行う	顧客が優先度を決定し限られた時間でデリバリを行う
WBSによる進捗管理によって管理される	機能のロードマップによって管理される
トップダウンで指示及び制御される	自立的に活動するチーム同士が協同する
しっかりと規定された重い方法論を使用する	必要最低限の軽い方法論を使用する
官僚的な評価基準とコントロール	価値に着目した評価基準

ソフトウェア開発プロジェクトの成功率の比較



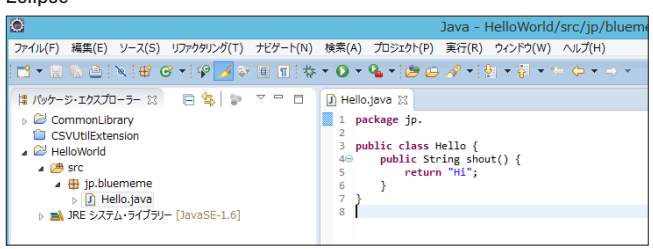
開発ツール「Service Studio」の基本情報

「Application」

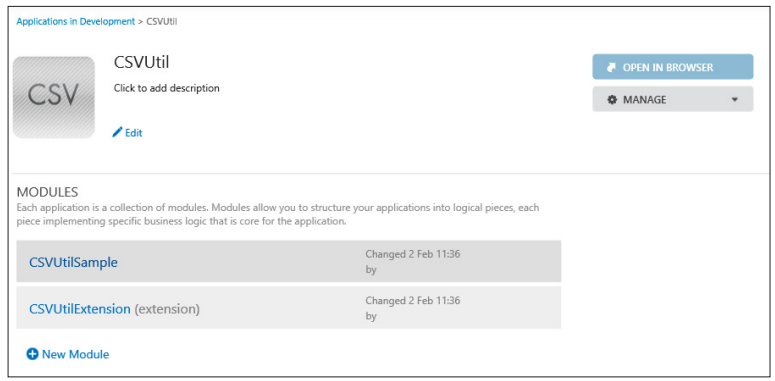
「Application」とは.NETのソリューションやJavaのワークスペースのようなものです。「Application」は、０個以上の「Module」から構成されます。「Module」には2種類あり、1つ目は「eSpace」、2つ目は「Extension」です。「eSpace」は「Process」、「Interface」、「Logic」、「Data」から構成されます。「Extension」は、「eSpace」の機能を拡張させるためのものです。「Action」、「Entity」、「Structure」を「eSpace」に追加することができます。通常、「Module」は特定の「Application」に紐づけられますが、特定の「Application」に紐づかない「Module」は「Independent Modules」と呼ばれる特殊な「Application」に表示されるようになります。「Independent Modules」は環境間のPublish作業によるミスを誘発するため、可能な限り避けて下さい。

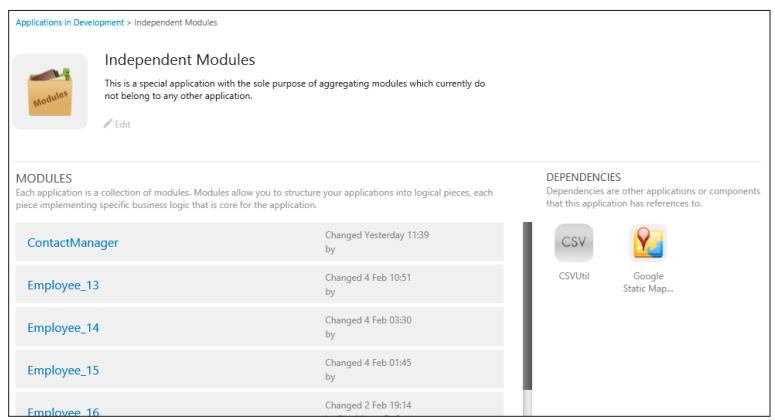
Visual Studio

Eclipse



Service Studio

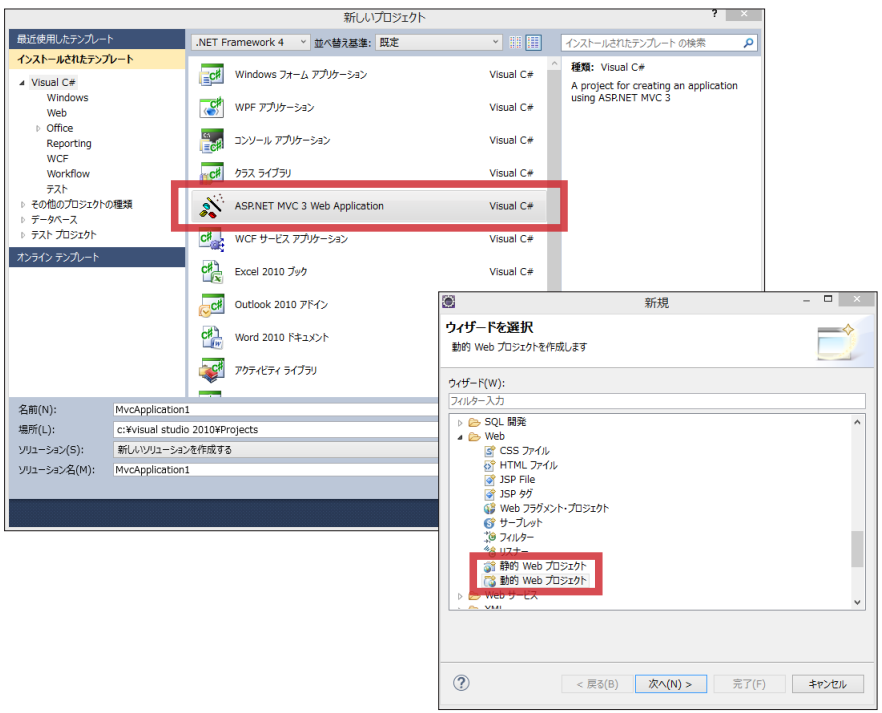




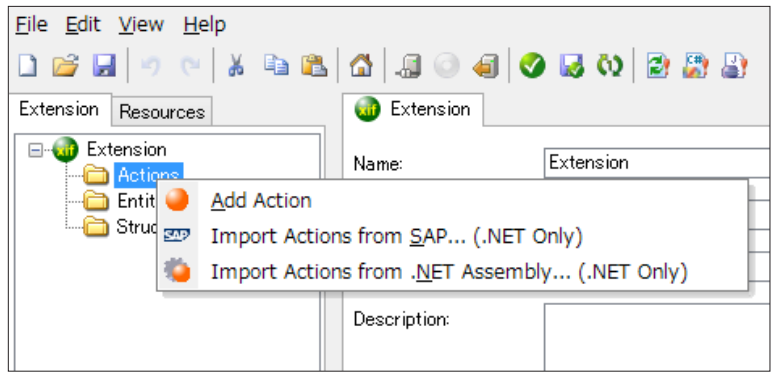
「eSpace」と「Extension」

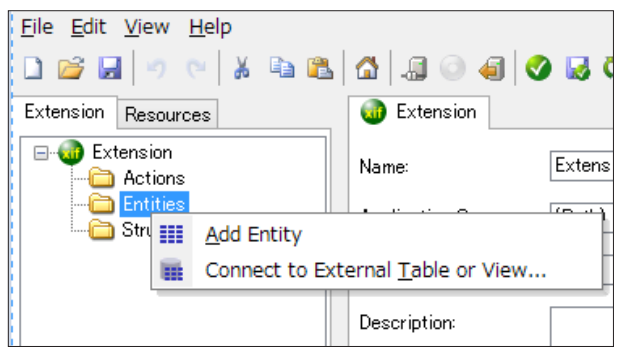
「eSpace」は.NETではWebアプリケーションプロジェクト、JavaではWebプロジェクトにあたるものです。OutSystems Platformでは「eSpace」と呼びます。「Extension」は外部システムとの連携、既存データベース／ライブラリの再利用、OutSystems Platformでは提供されていないロジックを追加することができます。

Visual Studio（上）、Eclipse（下）



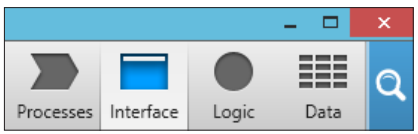
Integration Studio





「eSpace」の設計方法

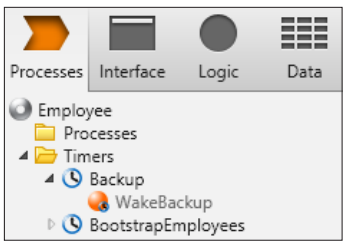
「eSpace」は４つのモデルを設計しながらアプリケーションを作成します。



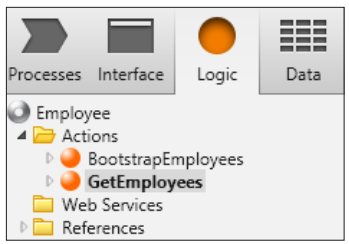
- **Processesレイヤ**：
  請求処理・注文処理・クレーム処理などのビジネスプロセスを設計したいときに使用します。
- **Interfaceレイヤ**：
  画面設計を行いたいときに使用します。
- **Logicレイヤ**：
  データベースへのアクセスや計算処理などを設計したいときに使用します。
- **Dataレイヤ**：
  データベースのテーブル設計などを設計したいときに使用します。

一覧画面を作成したいような場合でも、ブラウザに表示する画面はInterfaceレイヤ、テーブルからデータを取得する場合はLogicレイヤ、一覧に表示するためのデータの格納方法を設計する場合はDataレイヤ、夜間にテーブルに格納されたデータをバックアップする処理を設計するにはProcessesレイヤといったように、複数のレイヤを切り替えてアプリケーションを作っていきます。

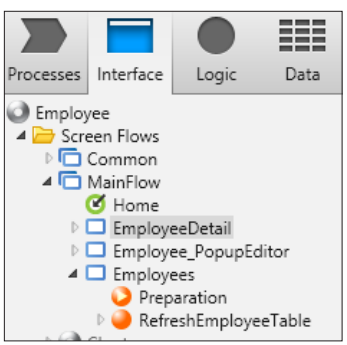
Processesレイヤ



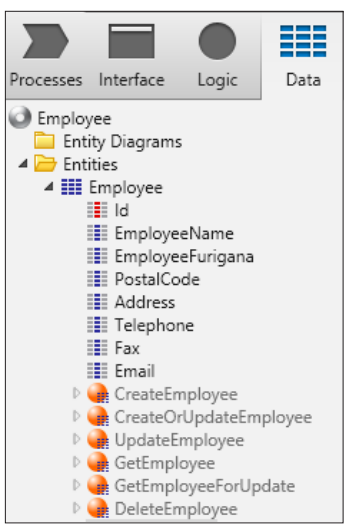
Logicレイヤ



Interfaceレイヤ

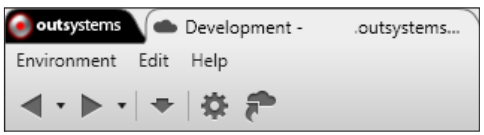


Dataレイヤ



Application管理画面のメニュー

左から2番目のタブで「Application」を管理することができます。このタブでは、「Application」の一覧・編集・削除・新規作成、「Module」の移動、「OutSystems Forge」から様々な「App lication」や「Module」をインストールすることができます。画面左上のメニューには5個のアイコンが表示されています。



左から順に「画面を戻る」、「画面を進む」、「サーバから開く」、「管理コンソールへの接続」、「他のPlatform Serverに接続」の機能が並んでいます。

- **画面を戻る**：
  直前に開いていた画面に戻ることができます。「Application」を開いている状態から、「Application」の一覧画面に戻ることができます。
- **画面を進む**：
  画面を戻るで行った画面遷移をやり直すことができます。
- **サーバから開く**：
  「eSpace」のバージョンを指定して開くことができます。Publishされているバージョン以外を開きたいときに利用しますが、利用頻度は低いです。
- **管理コンソールへの接続**：
  Service Centerと呼ばれるPlatform Serverを管理するためのコンソールを表示することができます。アプリケーションでエラーが発生したのでエラーログを確認したいといった時によく使います。
- **他のPlatform Serverに接続**：
  Platform Serverがインストールされている環境が複数ある場合に利用します。現在接続しているPlatform Serverがない場合は、「CONNECT TO ENVIRONMENT」をクリックしたときと同様の動作をします。現在接続しているPlatform Serverから、他のPlatform Serverへ接続を行うことができ、接続しているPlatform Serverの数だけService Studioが起動することになります。

Application管理画面の操作方法

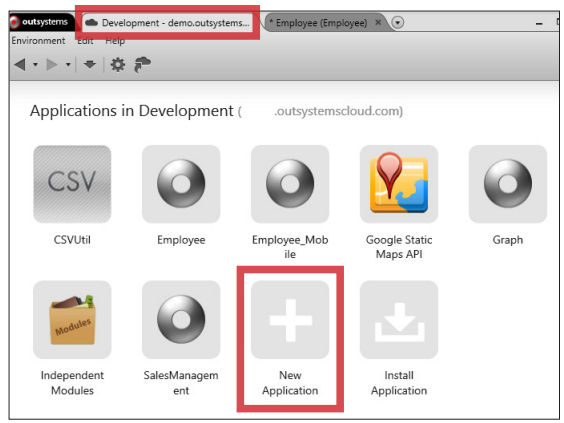
01 左から2番目のタブを開き、Applicationの一覧を開く。

- Applicationの詳細が開かれている場合は画面の戻るアイコンをクリックするか、詳細の左上に表示されているパンくずリストから「Applications in Development」のリンクをクリックし、Applicationの一覧画面を表示する。

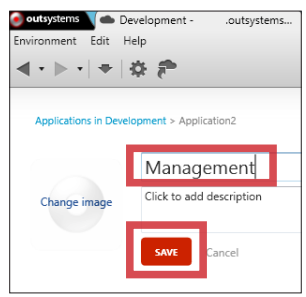
02 「New Application」をクリックする。

03 「Application1」を「Management」にリネームし、「SAVE」をクリックする。

01・02



03

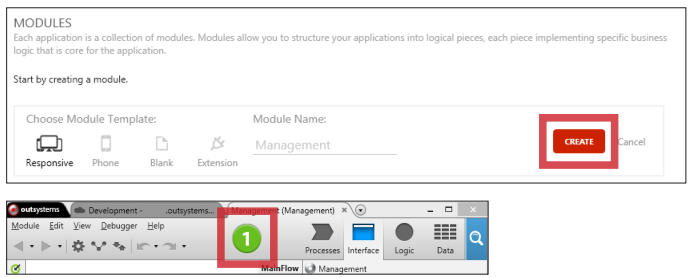


04 「CREATE」をクリックし、緑色の①ボタンをクリックし、1-Click Publishを行い、青の矢印ボタンに変わるまで待つ。

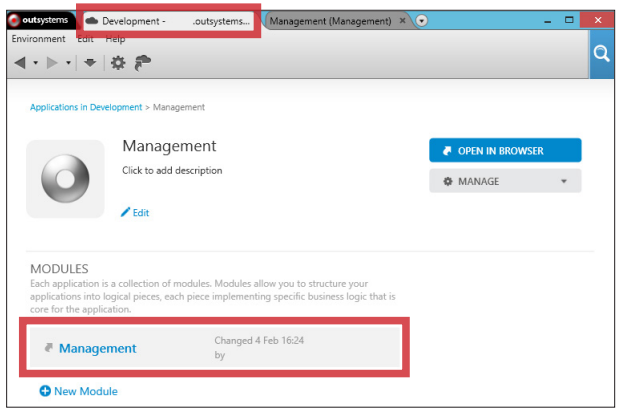
05 左から2番目のタブをクリック。

- 作成した「Management」eSpaceが表示されます。

04



05

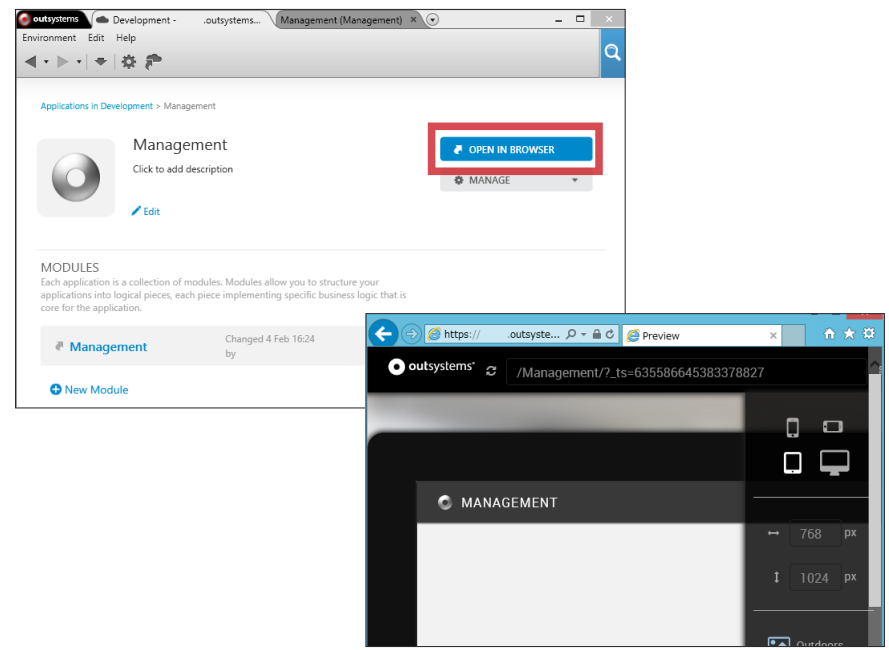


「OPEN IN BROWSER」はHome設定されているeSpaceをプレビューモードで開くことができます。Home設定は「Application」内で一つだけ設定することができますが、ライブラリだけをまとめた「Application」の場合、Home設定を行わないこともできます。

06 「OPEN IN BROWSER」をクリックすると、ブラウザが起動する。

- 「Management」eSpaceは画面設計が初期状態なのでプレビュー画面内にヘッダとフッタだけが表示されます。

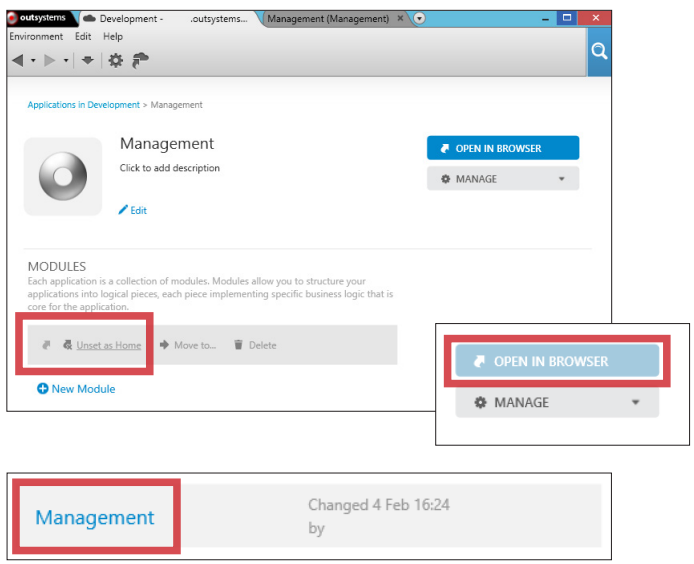
06



07 「Management」eSpaceにフォーカスし、「Unset as Home」をクリックする。

- 「OPEN IN BROWSER」がグレーアウトし、クリックできなくなります。
- 「Management」eSpaceの先頭の矢印が消えます。

07

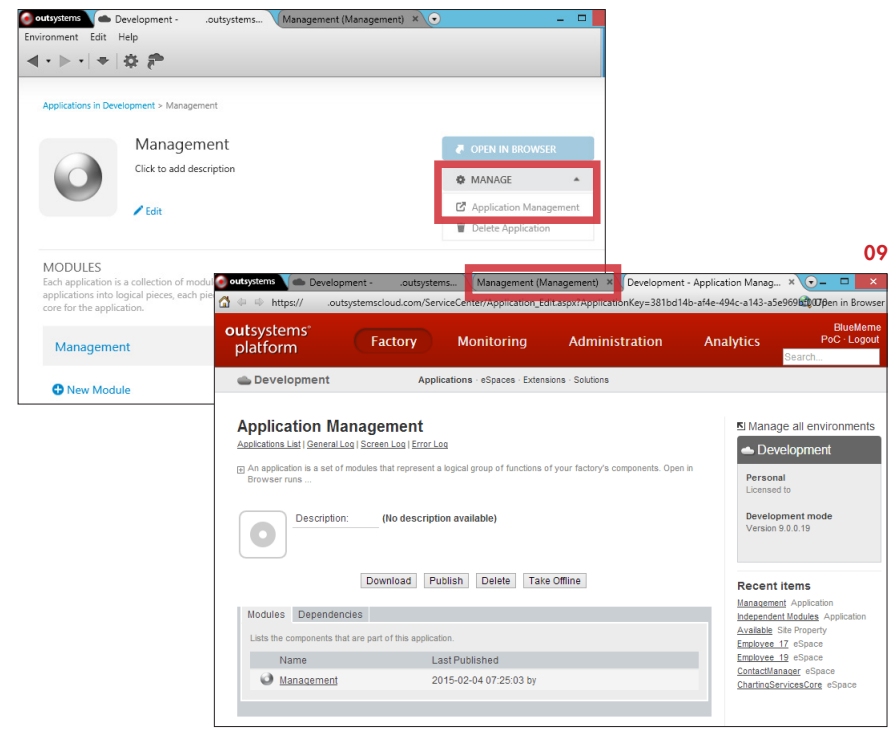


08 「MANAGE」をクリックし、「Application Management」をクリックする。

- Service Centerが開き、「Management」Applicationの管理画面が表示されます。

09 確認後、左から2番目のタブをクリックする。

08

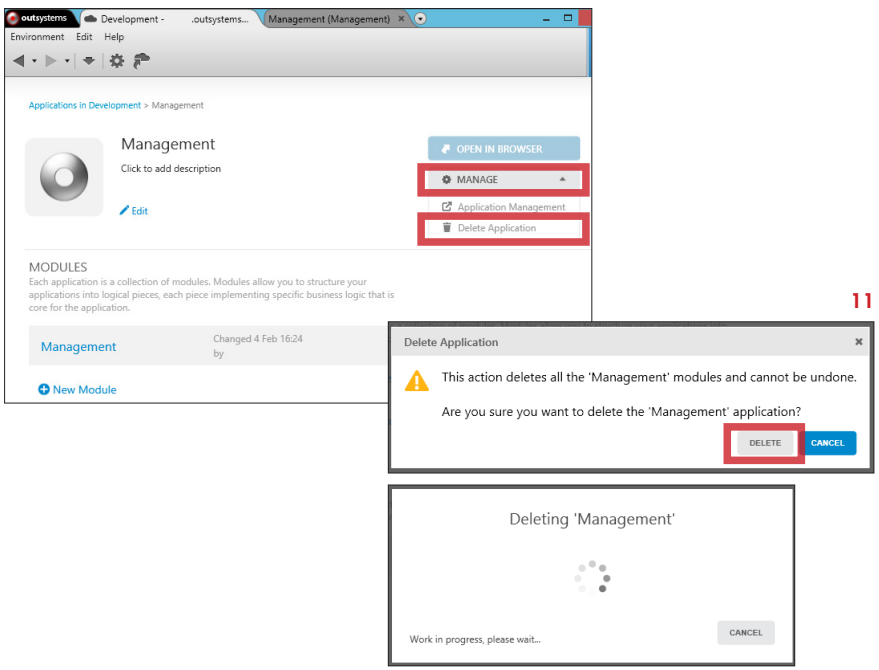


10 「MANAGE」をクリックし、「Delete Application」をクリックする。

11 「DELETE」をクリックする。

- 「Application」一覧画面に自動遷移し、「Management」Applicationが消えています。

10



11

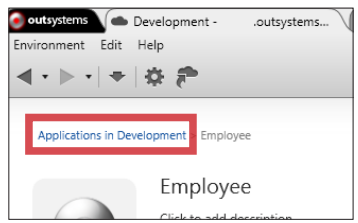
Independent Modulesからの Moduleの移動

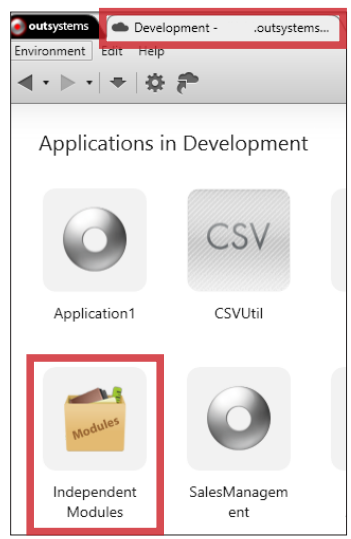
01 左から2番目のタブを開き、Applicationの一覧を開く。

- Applicationの詳細が開かれている場合は画面の戻るアイコンをクリックするか、詳細の左上に表示されているパンくずリストから「Applications in Development」のリンクをクリックし、Applicationの一覧画面を表示する。

02 「Independent Modules」をクリックする。

01

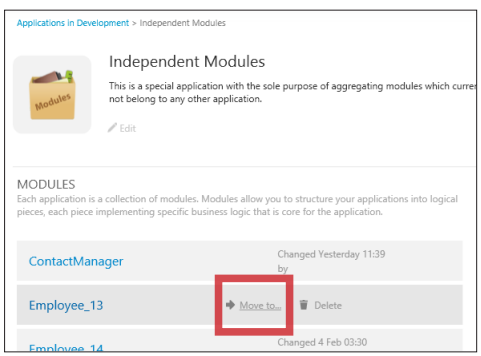




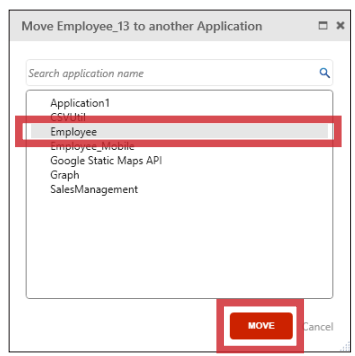
03 移動したい「Module」にフォーカスすると「Move to...」、「Delete」の項目が表示されるので「Move to...」をクリックする

04 移動先のApplicationを選択する。

03

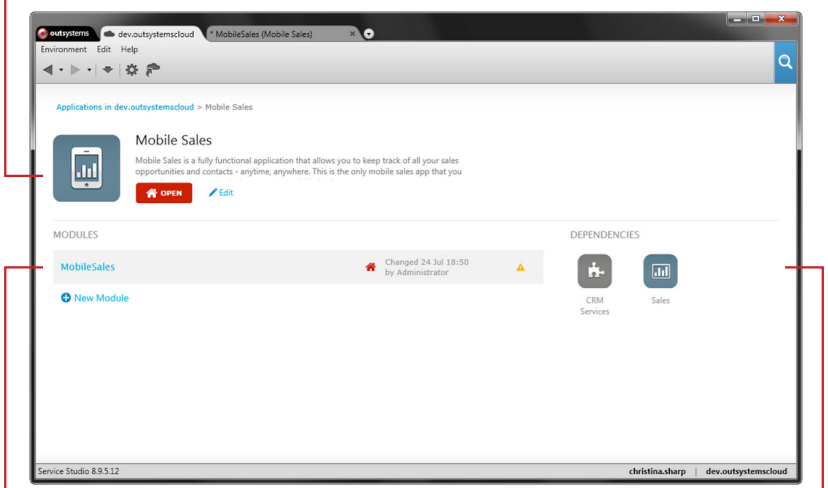


04



Service Studioの各種名称

Application概要



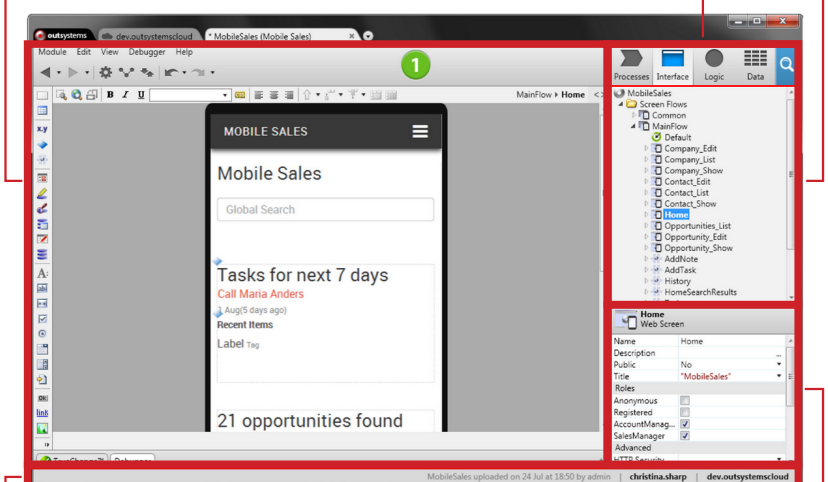
Application Module一覧

依存Application一覧

エディターエリア

エレメント

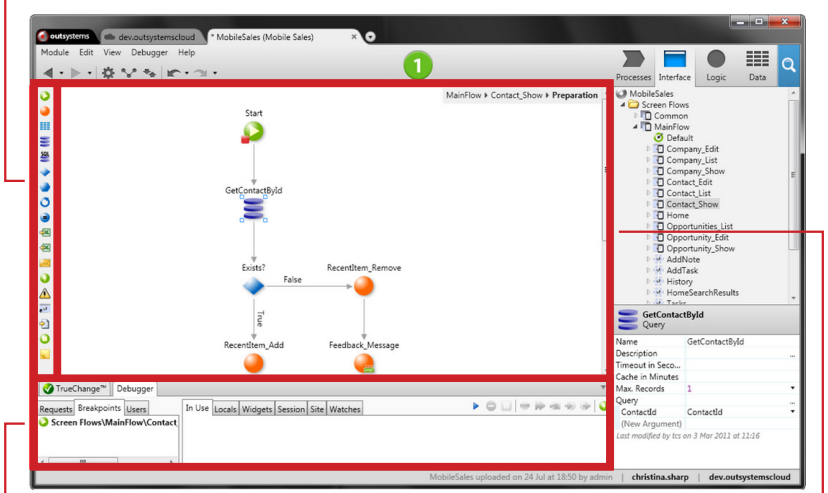
レイヤー



ステータスバー

プロパティペイン

ツールボックス

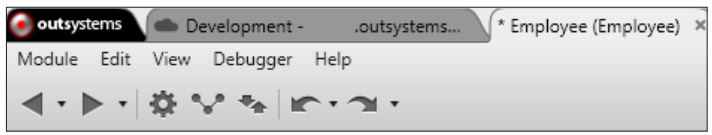


デバッグエリア

コンテントキャンバス

「eSpace」の設計画面でのメニュー

左から3番目以降のタブが開かれているときがeSpaceの設計画面になります。画面左上のメニューには7個のアイコンが表示されています。



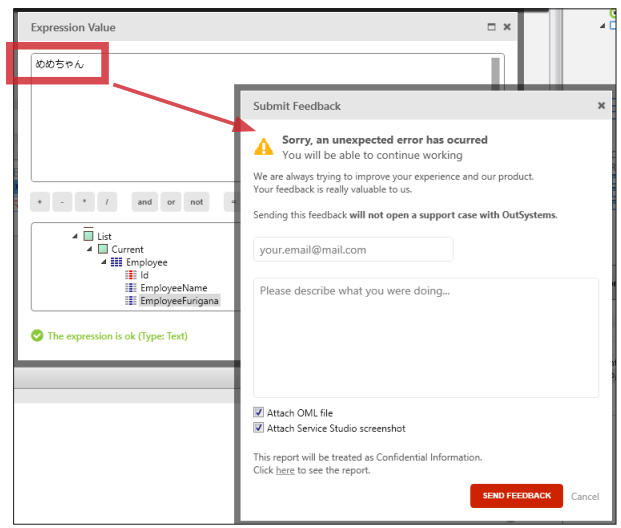
左から順に「画面を戻る」、「画面を進む」、「管理コンソールへの接続」、「参照の追加と削除」、「eSpace同士の比較」、操作を元に戻すための「Undo」、操作をやり直すための「Redo」の機能が並んでいます。

- **画面を戻る**：
  直前に開いていた画面に戻ることができます。Interfaceレイヤの詳細画面を開く前に、Logicレイヤのバックアップアクションを開いていれば、レイヤ間を跨いでも直前に開いていたバックアップアクションを開くことができます。
- **画面を進む**：
  画面を戻るで行った画面遷移をやり直すことができます。
- **管理コンソールへの接続**：
  Service Centerと呼ばれるPlatform Serverを管理するためのコンソールを表示することができます。アプリケーションでエラーが発生したのでエラーログを確認したいといった時に、よく使います。
- **参照の追加と削除**：
  「Extension」の機能を取り込んだり、他の「eSpace」の画面やテーブル情報などを利用できるようにします。
- **eSpace同士の比較**：
  自分の設計している「eSpace」とサーバ上にPublishされている「eSpace」の差分比較を行います。差分がある場合は、グラフィカルな画面で差分を確認できます。
- **Undo**：
  直前の操作をなかったことにできます。画面のボタンを削除してしまったけど削除するのはなしにしたい、設計情報を修正したはいいけど変更前の設計情報に戻したい、といった時に活用します。
- **Redo**：
  Undoでの操作を取り消すことができます。Undoしすぎてしまった時や、どういう変更を加えたかを再確認したい、といった時に活用します。

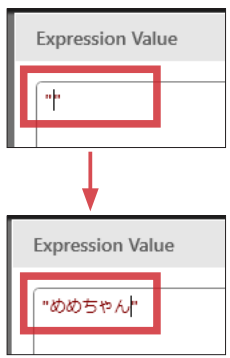
Service Studioを操作する上での注意点

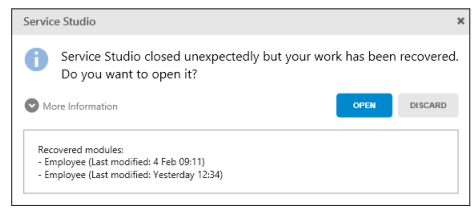
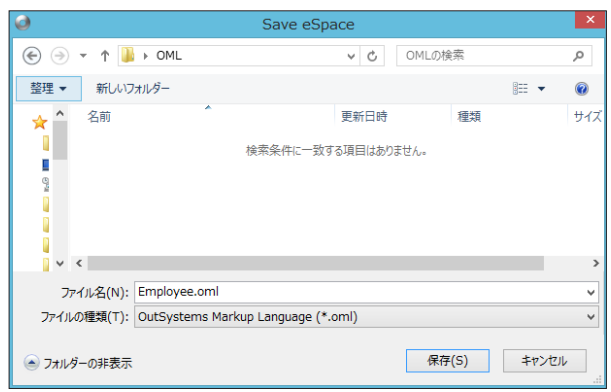
定数文字列を入力する際は「"（半角ダブルクオート）」を先に入力します。Expression、Property、Aggregateなどで全角文字を入力する際に、半角ダブルクオートを先に入力する癖をつけておくと、不意のクラッシュを避けることができます。

誤り例



正しい例



自動保存

突然Service Studioが終了してしまった場合に備え、自動保存の機能があります。次回、Service Studioを起動すると、自動保存されている「eSpace」を開く旨のというダイアログが表示されます。自動保存された「eSpace」を開く場合は「OPEN」を、開かない場合は「DISCARD」をクリックします。自動保存の機能もありますが、適宜、Ctrl+Sで「eSpace」をファイルに保存しておくことを推奨します。「1-Click Publish」が行えないような状態が続いたり、開発PCの不調によりService Studioが終了してしまった場合でも、復旧することが容易です。