CTL Customer Services Solution Overview

Aptean Ltd

Contents

1 CTL Customer Services Solution Overview

This is a technical overview of the CTL project as used for customer services and fleet management.

1.1 Projects

- CTL-TMS - UI - webforms, JQuery, etc
- TMSCustomServerControls - UI - reusable standardised controls
- TMSDatabaseClassLibrary - Database access - DAL, BAL, SQL, models
- TMSDatabaseScriptLibrary
- TMSResourceClassLibrary - Multi-lingual, labels, messages, etc.
- TMSUtilityClassLibrary

1.1.1 CTL-TMS

The main UI.

- App_Data - a list of text entries that should not be allowed. Security, prevents trojan attacks.
- App_Start - Project folder
- bin - included .NET objects. DLLs generally.
- Content - reusable aspx/aspx.cs and JS content. Think include files for commonly used things.
    - Internal - reusable items that we have written. reusable aspx/aspx.cs. Lots are to do with CTL only, but some are super-important - called out below. Usually included in teh Master page that the webform page is based on.
        - Toolbar - used to define the toolbar
        - AuditTrailGV - CTL - N/A
        - Calender - CTL - N/A
        - Clock - CTL - N/A
        - ContactsMaint
        - CustomerService specific to customer services screen
        - CustomServerControls
        - Footer - generic footer for every screen.
        - HereMap
        - LeftSlider - left pop-out pane
        - LocationAssociation - CTL - N/A
        - LookupList
        - MenuSlider
        - NoteMaint
        - OverheadDataFeed
        - ParameterDetail - CTL - N/A
        - ReportHub - specific to report hub
        - Requirements
        - RightSlider - right pop-out pane
- Controllersempty - usually used for MVC forms. There are none in the customer services CTL at the moment, several (but not many) in CTL project.
- DataFiles - a list of text entries that should not be allowed. Security, prevents trojan attacks.
- gateway - unsure - think might be to do with Customer Servies Screen processing.
- Images - images used by CSS and webforms. E.g. the Aptean logo.
- Models - empty - usually used for MVC forms. There are none in the customer services CTL at the moment, several (but not many) in CTL project.
- obj - built objects
- Properties - project properties
- Scripts - Javascript included in webforms - reusable browser-based code, commonly written by someone else. E.g. all the JQuery stuff is in here.
- Service References
- Skins - JQueryUI schemes, CSS
- WebForm - Where the forms front-end (.aspx) and back-end/server (.aspx.cs) code exists
- WebServices - CTL - N/A

In the top level are also the master forms. think of these as the basic template of every form that references it.

- AdminSite.Master - most screens inherit this
- Basic.Master
- Blank.Master

- EndGateway.Master
- MainOnly.Master
- Planning.Master
- Site.Mobile.Master - doesn't seem used.

There are also some basic HTTP Error forms that ae defined here:

- Errors.htm
- InvalidAccess.htm
- LoggedOut.htm
- NotAuthorised.htm
- SessionExpired.htm

Theere are also some others:

- Default.aspx - a default form for the project - N/A
- ViewSwitcher.ascx - something to do with friendly URLs and mobile views. Doesn't seem used by anything except the site mobile master, which itself seems unused.

1.1.2 TMSCustomServerControls

UI - reusable standardised controls

- ButtonControls - defines classes for all common buttons that might be used in the application, along with the styles that are applicable to the button in that area
    - FA - Slider Menus
    - AT - Table buttons
    - FT - Footer buttons
    - Others
- CheckBoxControls
- DropDownListControls - defines the drop-down list PLUS the data table access required to populate that drop-down list. There are dozens of these.
- GridViewControls - controls related to the main gridview
- HyperLinkControls - don't appear to be used in this project
- LabelControls - used everywhere for labels of any kind, in grid views, accordions, etc.
- LiternalControls - not used
- PageControls - reusable button click events, attached to control events, e.g. toolbar buttons
- RadioButtonControls
- TextBoxControls - All types of data entry, from generic reusable types to specific field types. You can define lookups in here and it will add the buttons for you. The lookups are defined in the LookupBAL/DAL objects. Assume that every unique field will have a custom TextBox control defined here. Most will simply reuse a core class such as
    - ..General
    - ..Filter
    - ..Alpha
    - ..Website
    - ..Numeric
    - ..Integer
    - ..Percentage
    - ..Decimal
    - ..etc
    - TextBoxProductCode builds on TextBoxAlpha, which itself builds on TextBoxGeneral, which builds on TextBox, which is the .NET basic text box control.

1.1.3 TMSDatabaseClassLibrary

- BAL = Business Access Layer = Business Logic
- DAL = Data Access Layer = the core components to select, update,insert on table(s) in this DAL object, including all variants.

Naming convention varies, but generally after the table or view name (and in a lot of cases, after the CTL table name).

Typically every table will need a

- Model

    - A simple, usually generated
    - A more complex one listing the fields and any properties and types.
- DAL
    - Again, in pairs, this defines the selects required, inserts, updates, etc.
- BAL
    - Some objects require BALs. This typically collects properties nd methods relating to this data object above and beyond the simple SQL functions. For example, the BAL may contain static lists or classes that define the data that we would use, or functions relating to the object, like calculating collection and delibery windows for an order.
    - Predominantly, BAL business logic functionality will not exist - this will remain in the database, but there may be functions to CALL these BL procedures from within .NET within the BAL.

1.1.4 TMSDatabaseScriptLibrary

Database Change scripts and customer configuration scripts. part of the release process. Probably superseded by Carl's CTMS release process.

1.1.5 TMSResourceClassLibrary

Multi-lingual, labels, messages, etc.

- AppMessages - messages
- AppTerms - labels

1.1.6 TMSUtilityClassLibrary

Lots of really general (and some really specific) utility classes. Useful for coding/reusability.

- WebService.cs
- Exceptions
- obj
- Properties
- Resources
- Callback.cs
- Cookies.cs
- CustomEvents.cs - HTML Error event logging - related to hosting log files
- DataProcess.cs - specific to CTL Import process - N/A
- Exception_Handler.cs - handing exceptions - seems to use the Exceptions filder, and seems to be specific to CTL
- General.cs - coding utility classes. Like how to generate a random numner.
- MCSUtilities.cs - TimeZone functions. Not used anywhere.
- Message.cs - functions to take standard messages (in the Resource classes) and insert values at placeholder positions. Includes some pre/suffixes to help identify what type of message it is and therefore the formatting. Does not seem used.
- MessageDuration.cs - used when popping up messages (Content/Internal/Toolbar)
- Security.cs - used mostly in adm01_Login
- Validation.cs - common validations. Used in CustomServerControls like TextBox, GridView, etc.