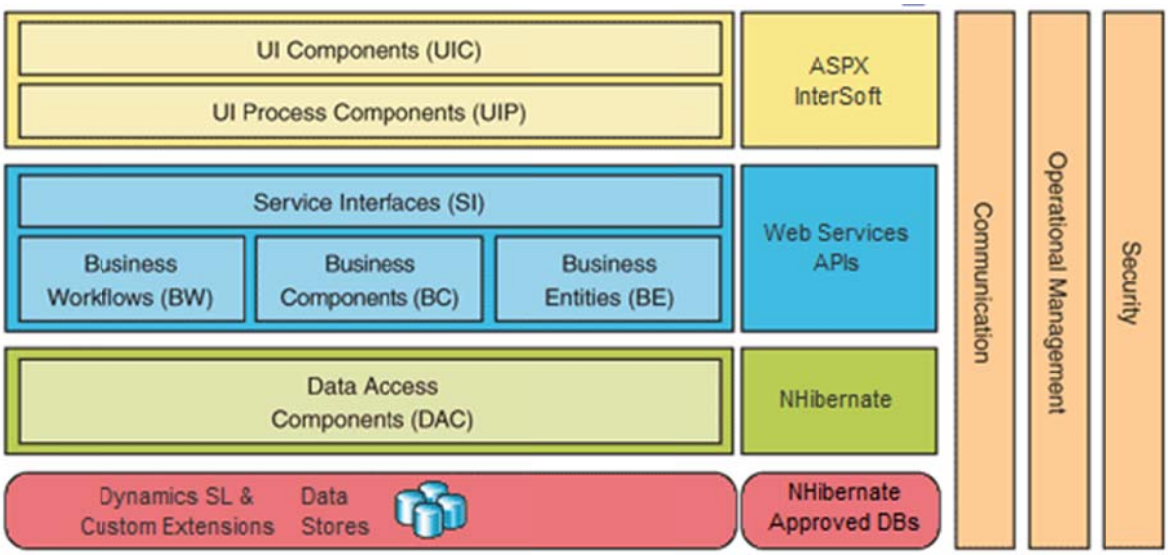
Web Development Framework

empower Provides a Web Development Framework designed as a Three-Layered Services application following Microsoft's MSDN patterns and practices.

An Entity Builder enables users to modify existing Entities such as Project, Customer, etc. or to build their own custom Entity unrelated to SL. An Entity is a collection of database objects (tables or views) and business logic compiled into an API. These APIs are consumed by the empOwer Enterprise Management Portal or can be published as separate Web Services.

The empOwer Web Form Editor and Views Editor are provided for creating or modifying Web Forms and Web Views (data lists) to enable data viewing and maintenance in the Enterprise Management Portal.



The above schematic shows a representation of the layers and technologies used within the empOwer application including; Presentation, Business and Data.

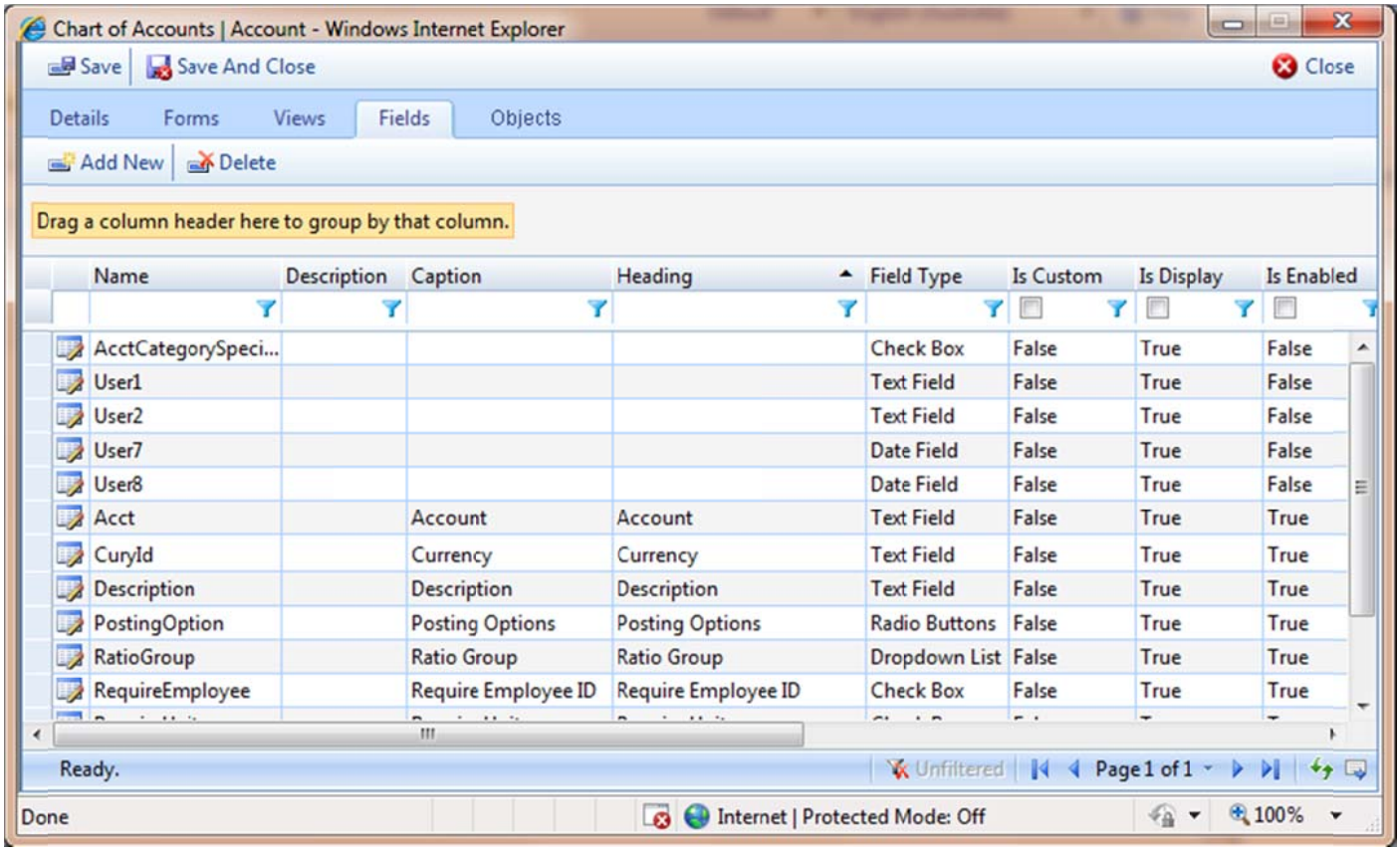
- **Presentation Layer** - provides the Web User Interface. EmpOwer has been developed using ASP.NET and InterSoft web controls.
- **Business Logic Layer** - applies the business functionality and controls required for data validation, error handling and data processing. The Dynamics SL Entities apply similar business logic to the SL screens.
- **Data Access Layer** - interfaces to the databases using NHibernate and can integrate to any database supported by NHibernate.

Entity Builder

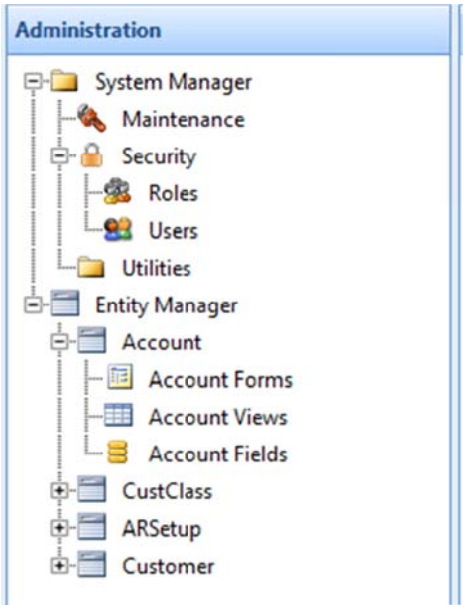
The Entity Builder allows users to select an underlying database object, or another existing Entity, as the foundation to construct a new Entity. It is then possible to attach further database objects or Entities on either a 1:1 or one to many basis. The object's fields and their properties are displayed for user modification in the schema editor. Their properties default from the source Table/Entity.

Saving the Entity updates the Schema.xml file to be used by the Code Generator to build Entities.

Users can build multiple Web Forms or Views for each Entity enabling the Entity to be available for data entry in the Enterprise Management Portal. A Web Service can also be generated.



Administration



The Administration menu provides User and Role security management as well as the tools for maintaining, Entities, Views and Forms.

It is possible to define multiple Views and Forms for any Entity. The Entities, Views and Forms can be linked to security Roles limiting users access to input forms or data lists.

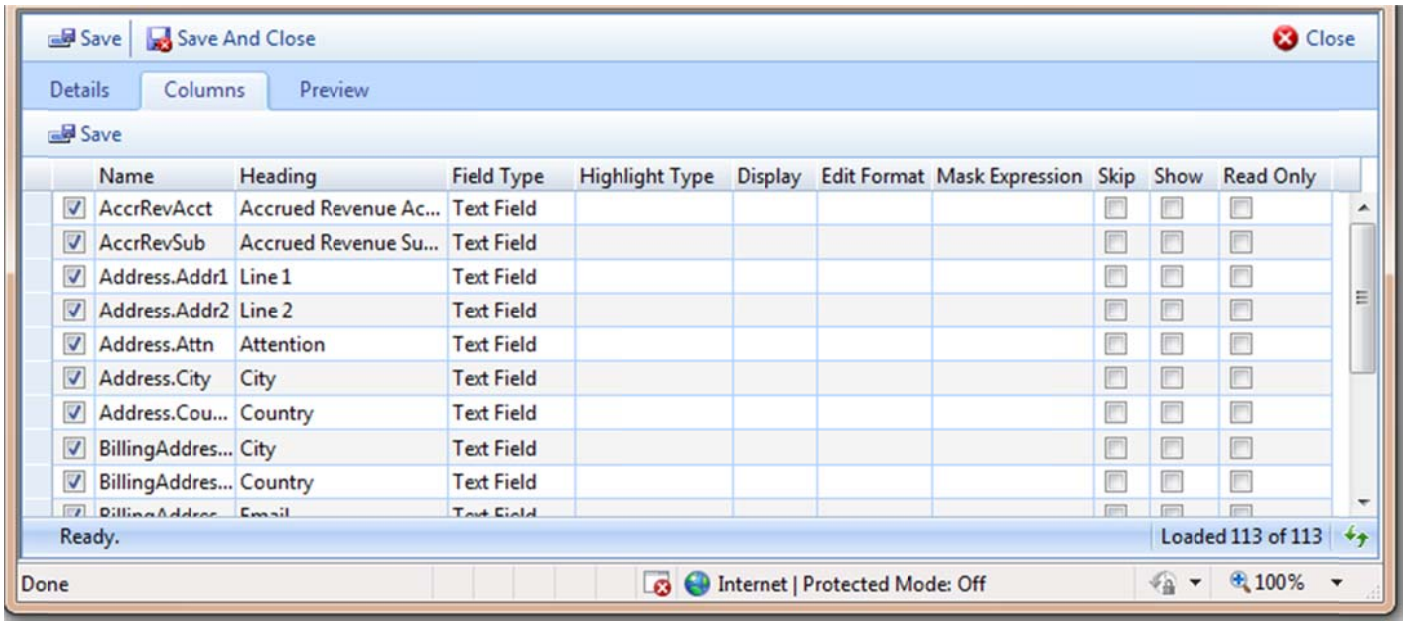
User specific Forms and Views can be defined to control any level of data access. For example specific display only Views can be created, or special forms hiding sensitive information.

Views can become a primary tool for managing the data users will have access to see, as well as enabling efficient filtering of records to improve search and retrieval processes for commonly used data sets.

View Builder [Grids | Data List]

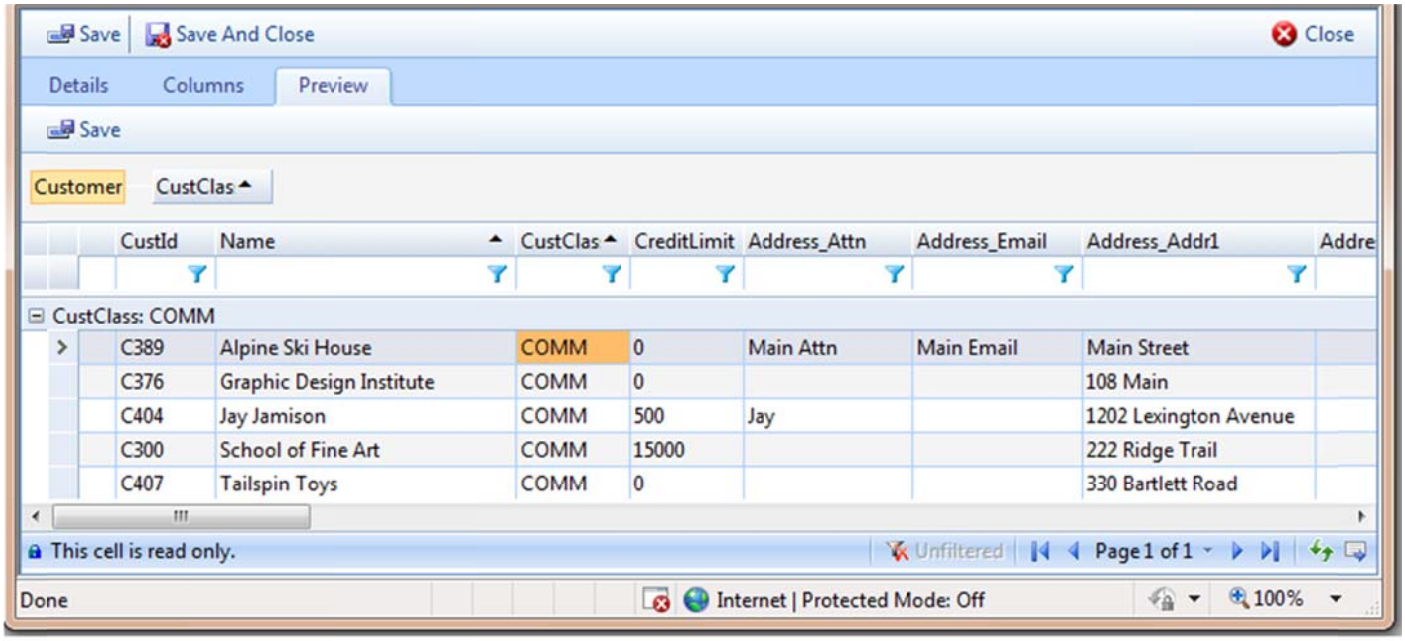
The View builder inherits all the fields and their properties or controls from the source Entity. The Entity may lock some of the field properties to prevent them being changed. For all remaining field properties the user can add extral controls or modify properties to restrict user access or improve data search, entry or maintenance.

Common properties include, whether the field will be displayed, editable or required, whether it will have format mask, be highlighted and its field type (dropdown, look-up or simple text).



The Details tab is used to store basic information for the View including, View ID, Name and the number of records that will be displayed in data list paging. This is the number of records displayed on each screen load or page scrolling / navigation.

Once the View and Field properties are defined the View can be previewed and Grouping, Sorting and/or default Filtering applied. Sorting can be used in limiting users access to specific groups of records, e.g. only customers in a specific sales territory.

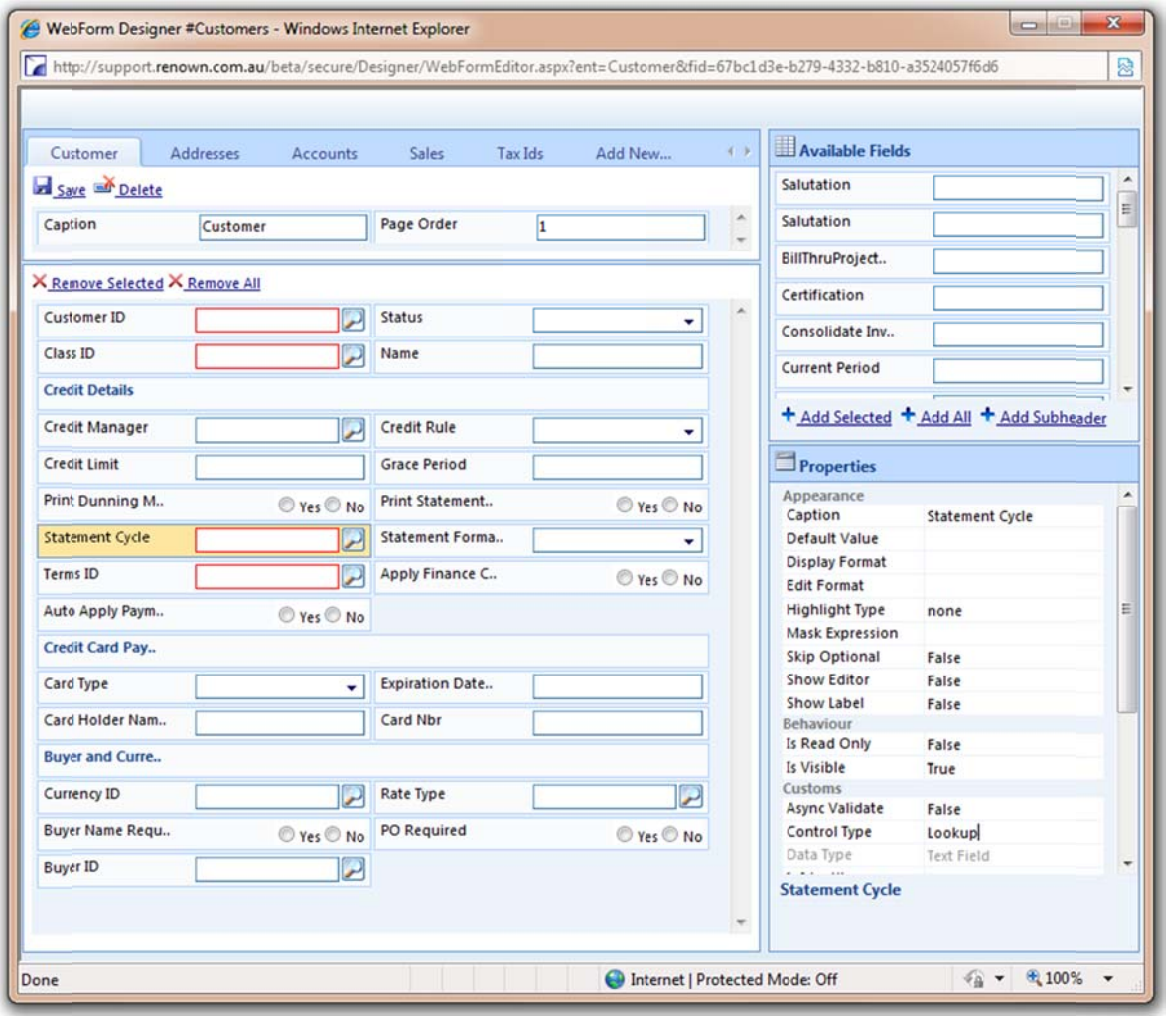


Web Form Designer

Multiple Web Forms can be created for each Entity. The Web Forms can be linked to a User Role and to Views. When an Entity is selected the main portal pane is populated with the data list based on the default view for the Entity. The selected View ID is displayed in the top line menu and can be changed to any other View the user has access rights to use.

The default form associated to the View is also displayed. If the user double clicks a row in the grid it is this Form that will be launched. The user can change the Form to any other they have access to prior to double clicking the grid, and the alternate selected Form will be launched.

The Form Designer inherits all the fields and their properties and controls from the Entity. Where Entity properties are set to fixed they cannot be modifed in the Form. It is possible to set different field properties between the Entity, View and Form, e.g. some fields may hidden in the grid, but displayed in the form.



Each Form can have multiple Tabs. The Entity fields can be added to each Tab in one of two columns. It is possible for the same field to be displayed on multiple Tabs if required. Headers can be used to seperate and group logical collections of fields.

Field properties can be defined to add controls, although most controls are likely to have been defined at the Entity level and inherited to the form. Fields can be configured as Textbox, Radio, Dropdown, Look-up, Date and many others.

Most if not all, field properties available in Customisation Manager in Dynamics SL will be provided in the property list of the Web Form Designer. Further controls and properties will continue to be added to the Designer as empower is developed.