**Smart CPQ**

Solution Design Document



Contents

[1 Introduction: Your Solution Design Document 4](#_Toc536044704)

[2 Solution Overview 4](#_Toc536044705)

[3 Project Information 5](#_Toc536044706)

[4 Workflows 5](#_Toc536044707)

[4.1 New Quote for a New Customer 5](#_Toc536044708)

[4.2 New Quote for an Existing Customer 6](#_Toc536044709)

[4.3 Process Diagram 6](#_Toc536044710)

[4.3.1 To-Be Process 6](#_Toc536044711)

[4.4 Process Illustration 6](#_Toc536044712)

[4.4.1 New Customer 6](#_Toc536044713)

[4.4.2 Existing Customer 7](#_Toc536044714)

[4.4.3 New PROS Quote 8](#_Toc536044715)

[4.4.4 Approval Process 10](#_Toc536044716)

[4.4.5 Proposal Generation 11](#_Toc536044717)

[5 Cart Design 13](#_Toc536044718)

[5.1 Spreadsheet Columns 13](#_Toc536044719)

[5.2 Top-Line and Left Sidebar Totals 14](#_Toc536044720)

[5.3 Header Fields 17](#_Toc536044721)

[5.4 OnDemand Grid Actions 21](#_Toc536044722)

[6 Product Catalog 22](#_Toc536044723)

[6.1 Catalog Structure 22](#_Toc536044724)

[6.2 Pricing and Discounts 24](#_Toc536044725)

[6.3 Business Properties 25](#_Toc536044726)

[7 Configuration Process 25](#_Toc536044727)

[7.1 Fire Risk Assessment 25](#_Toc536044728)

[7.2 Fire Training 26](#_Toc536044729)

[7.3 PFX 27](#_Toc536044730)

[7.4 Security 27](#_Toc536044731)

[7.5 Fire Systems 28](#_Toc536044732)

[8 Approval Workflow 29](#_Toc536044733)

[8.1 Approval Statuses 29](#_Toc536044734)

[8.2 Approval Rules 29](#_Toc536044735)

[8.2.1 Notification Panel 29](#_Toc536044736)

[9 Proposal Document 30](#_Toc536044737)

[9.1.1 Proposal Email 32](#_Toc536044738)

[10 Map-In and Map-Out 33](#_Toc536044739)

[11 [For PROS] Quick Start Implementation Files 34](#_Toc536044740)

[11.1 Excel Content File 34](#_Toc536044741)

[11.2 Theme/Layout Files 36](#_Toc536044742)

[11.3 Catalog Logo 37](#_Toc536044743)

[11.4 Settings Files 38](#_Toc536044744)

[11.5 cartServices.properties 39](#_Toc536044745)

[11.6 cartSettings.properties 41](#_Toc536044746)

[11.7 cameleon.properties 43](#_Toc536044747)

[11.8 Cart Model 44](#_Toc536044748)

[11.9 Jasper Report 48](#_Toc536044749)

[11.10 Map-In Fields 51](#_Toc536044750)

[11.11 Header Fields 52](#_Toc536044751)

[11.12 Preview Cart 54](#_Toc536044752)

[12 Appendix: Smart CPQ Catalog (CL) and Configuration Workflow (CP) Structure 56](#_Toc536044753)

[12.1 Catalogs 56](#_Toc536044754)

[12.2 Configuration Workflows 56](#_Toc536044755)

[12.3 Navigation 56](#_Toc536044756)

[12.4 Form Properties 57](#_Toc536044757)

[13 Version History 58](#_Toc536044758)

to do:

* configure Addresses header fields (three sets: SoldTo/BillTo/ShipTo; but only one set of Computed fields)
* quote statues (“Draft:Draft”)
* make quote uneditable when not in Draft

price method

integration.pros.pricing.pricingsvcurl = <http://wm-appsb1:8080/axis2/services/PricingService?wsdl>   
#integration.pros.pricing.pricingsvcurl = <http://wm-apptest1:8080/axis2/services/PricingService?wsdl>   
#integration.pros.pricing.pricingsvcurl =<https://sandbox.wm.proscloud.com/axis2/services/PricingService?wsdl>

# Introduction: Your Solution Design Document

This document is a high level solution design used as a starting point for configuration of PROS Smart CPQ. This document details overall configuration designs; **it is not intended** to document Chubb’s final go-live configuration, **nor is it intended** to document all aspects of configuration detail. Additional configuration details may be executed in the application User Interface (UI) and Solution Configuration File (SCF) that will not be documented here. Screenshots in this document are illustrative and meant to help define process. They should not be used to define exact configuration specifications.

This solution design document will ultimately be reviewed and signed off on by the Chubb project core team. After sign-off, any changes requested to this design may be subject to a change order at the discretion of the PROS Project Lead or Delivery Executive. Determination will be based on the effort to make the changes requested and adherence to project plan/metrics.

Through the Agile implementation methodology, additional configuration requirements may arise after each Sprint or Configuration Verification Workshops. At the discretion of the PROS project team, it may not be necessary to document some of the additional requirements in this solution design document. After solution design sign-off, configuration changes identified during configuration sprints will be documented in a configuration change log. Furthermore, these additional requirements may be subject to a change order based on an investigation led by the PROS Project Lead or Delivery Executive if such requirements are deemed out of scope based on the project Statement of Work (SOW).

# Solution Overview

PROS Smart CPQ extends your CRM [Customer Relationship Management] software to simplify the **configuration**, **pricing**, and **quoting** of products and services.

PROS Smart CPQ **Select-for-Sale** solution provides to your internal sales reps a streamlined quotation process for spot orders of standard products and integrated approval process based on quote score:

* Integrated catalog browsing
* Integrated quotation
* Integrated approval process based on quote score (if applicable) or other metrics defined in the cart
* PDF agreement document
* User interface with company branding

For Chubb, sales reps will use Smart CPQ within Microsoft Dynamics to browse a catalog of products. For the initial phase sales reps will only use CPQ/Microsoft Dynamics for the purposes of selling one of 5 services.

# Project Information

Chubb is fire and security provider of both equipment and services. This project will focus on an initial implementation of services only quoting in the UK. By implementing CPQ into their Microsoft Dynamics CRM, Chubb will modernize and digitize the quoting experience, shorten quote turnaround time, and enhance sales rep experience when quoting. The goal is to provide a MVP solution by end of March in time for the onboarding of 30 new sales reps.

# Workflows

There are two primary workflows for quoting a price agreement for services

* New Quote for a New Customer
* New Quote for an Existing Customer

## New Quote for a New Customer

The quoting process for a new customer/prospects begins by quoting directly off of a lead in Microsoft Dynamics. Sales reps will bypass account creation and checks using the existing Experian and MK Denial. Within the lead, sales rep will begin a new “PROS Quote”.

In the PROS Quote, the sales rep selects service(s) from a catalog and configures them based on the list of questions/options already defined on Chubb’s e-commerce customer facing portal and adds them to a cart. Within the cart the sales rep sees the price, any global discounts set centrally by Chubb for the services, and is allowed some discretion to provide additional discount. The system will not allow pricing if total discount for a service is below a hard floor. To finalize the quote the sales rep generates a quote agreement document which will be presented to the customer for signing either through invoice or through DocuSign.

(Note: For integration purposes, CRM opportunities will need to be created automatically once the PROS quote has been approved and payment received. This is due to the fact that the current integration with iSales is through the Dynamics opportunity. In order to streamline the quoting process, opportunities will not be required for creation of the PROS quote.)

## New Quote for an Existing Customer

The process for an existing customer is as same as for a new customer, but begins with finding and opening an existing account in Dynamics.

(Note: For integration purposes, CRM opportunities will need to be created automatically once the PROS quote has been approved and payment received. This is due to the fact that the current integration with iSales is through the Dynamics opportunity. In order to streamline the quoting process, opportunities will not be required for creation of the PROS quote.)

## Process Diagram

### To-Be Process



## Process Illustration

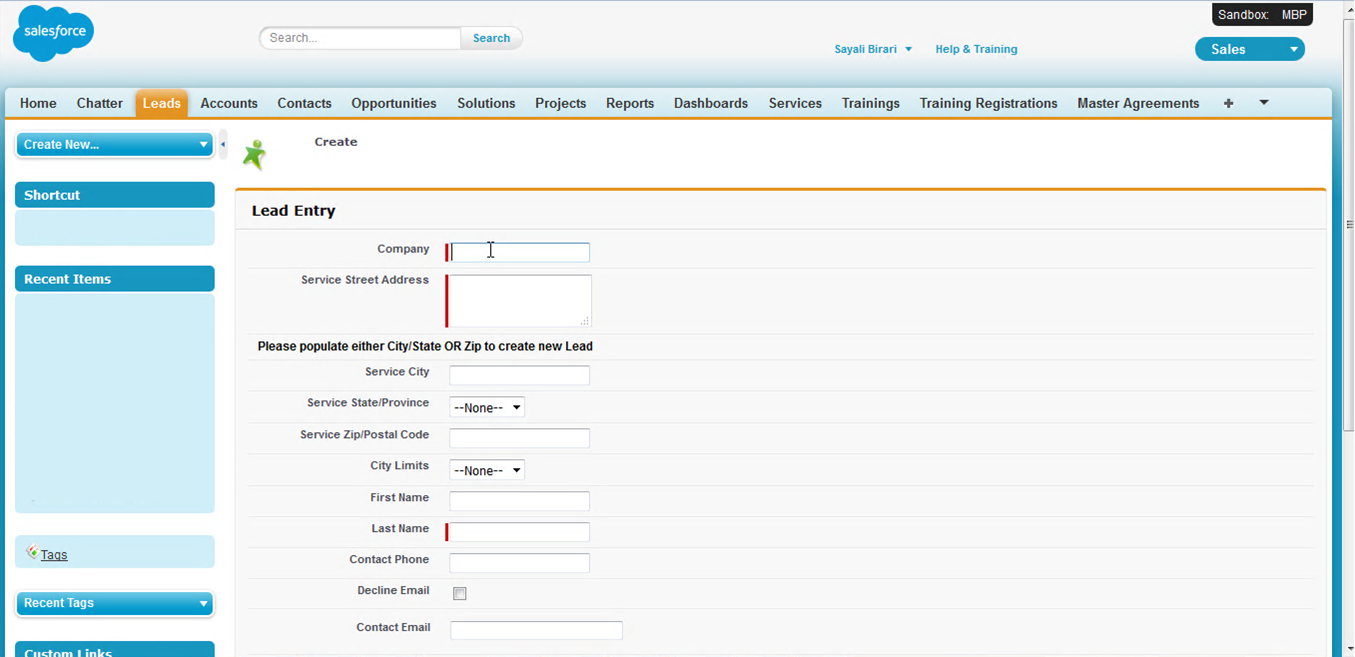
The following screenshots illustrate the above process diagram and the Smart CPQ **user interface** within {Salesforce, Dynamics}, including the **concepts** of:

* PROS Quote **cart**
* PROS Quote **catalog selection**
* PROS Quote **configuration process**
* Approval workflow
* Proposal generation

### New Customer

#### Create Dynamics Lead or Utilize Existing Lead

For new customers, a sales rep will create either a new lead or use an existing lead within Dynamics

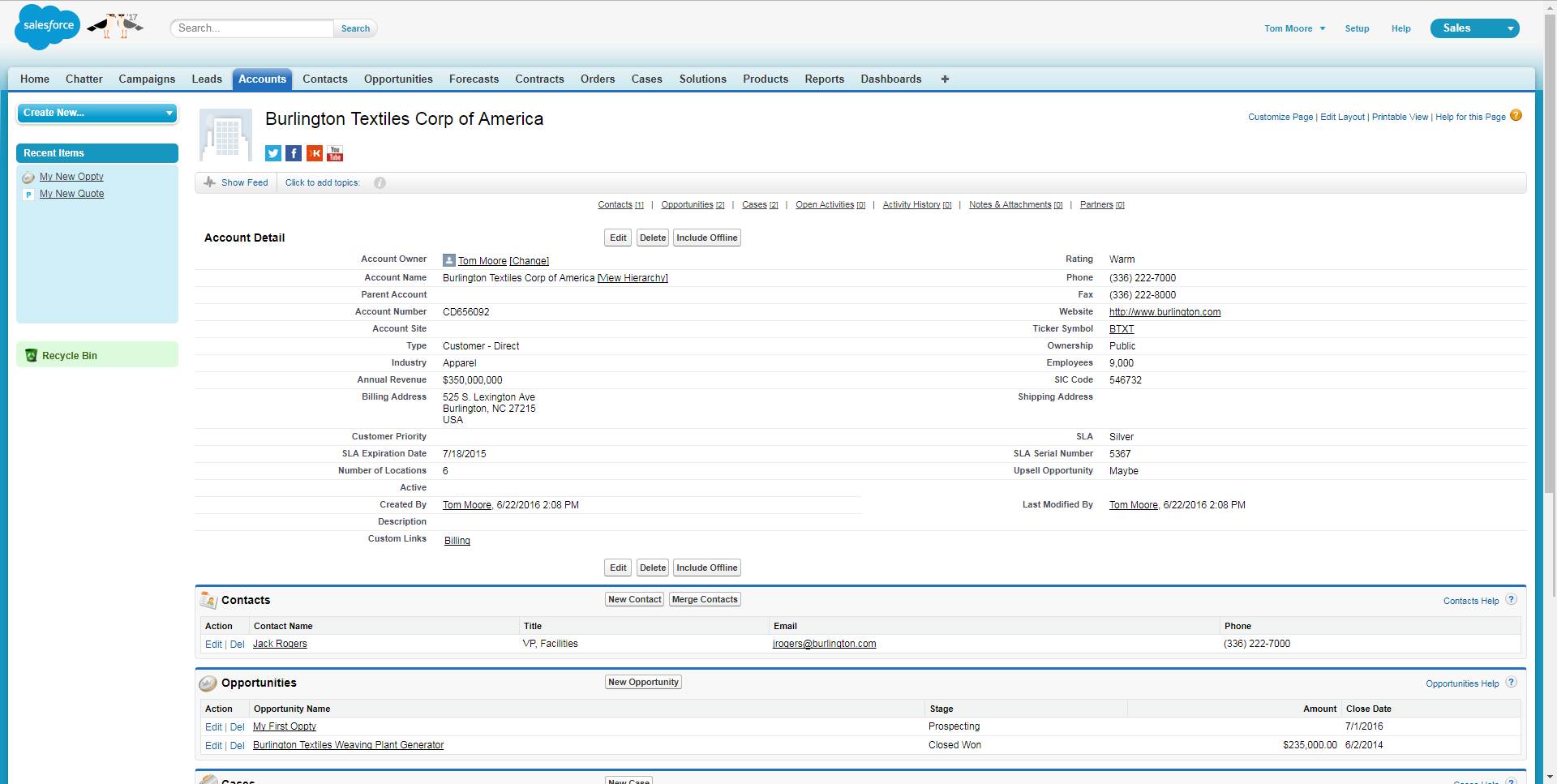


PROS will use the ‘Company Name’ of the lead as the entity with which to launch the PROS Quote from. Address information will need to reside within the Dynamics lead so that PROS can access it for document generation purposes

### Existing Customer

#### Open Dynamics Account

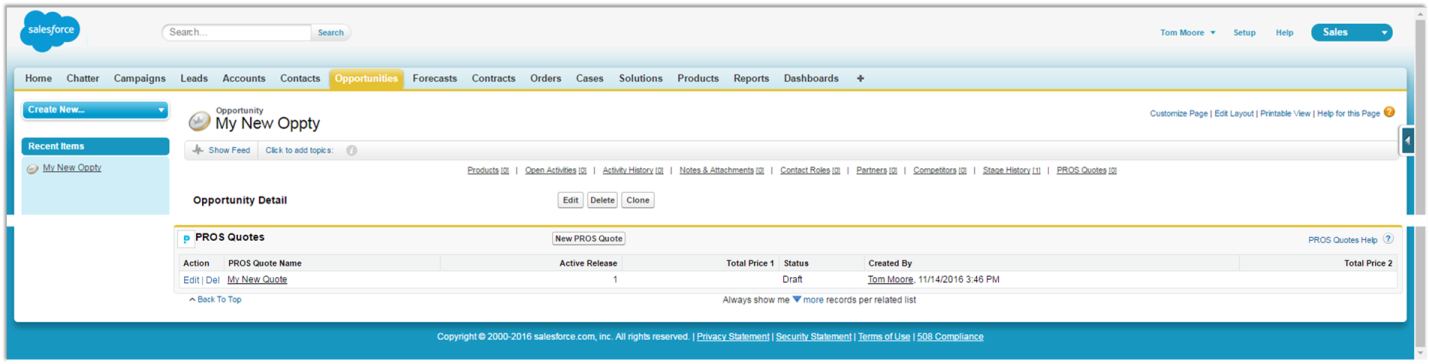
For existing customers, a sales rep will open an account either by account no. or by looking up the customer name, etc.



### New PROS Quote

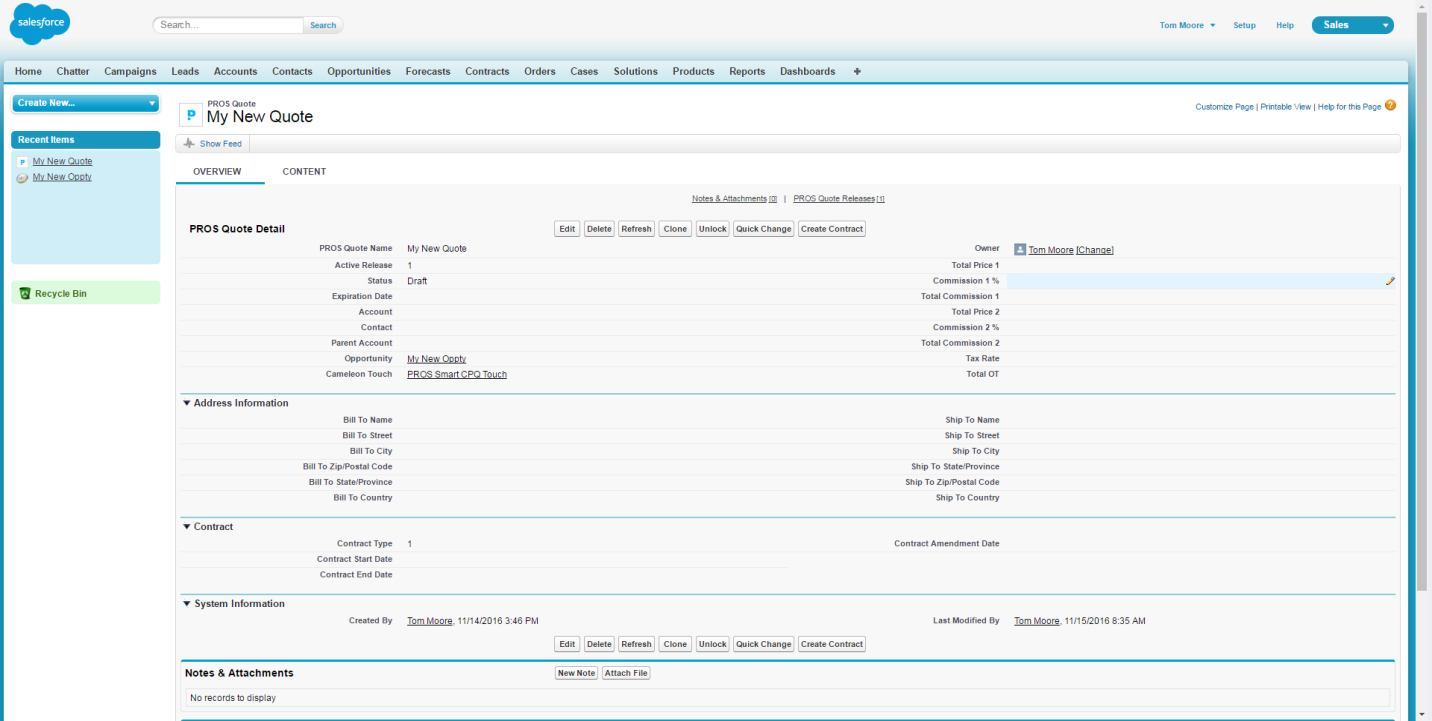
#### Create PROS Quote

For both new and existing customers, within the lead click “PROS Quote” to transition into a new Smart CPQ quote.



#### Overview Tab

“PROS Quote” opens on the PROS Quote “Overview” tab, which shows summary information from Dynamics.



Initially, this quote will only be populated with information that is readily available from the lead that the quote was initiated from. These fields will include:

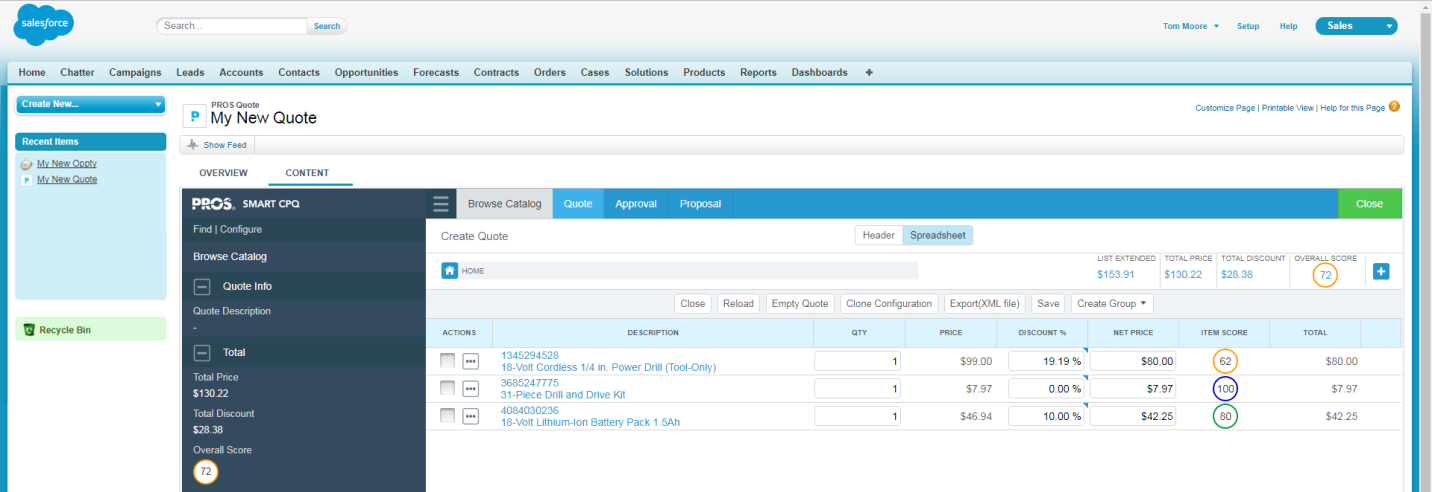
* Company Name
* Address
  + Street
  + City
  + Postcode

#### Content Tab

The Content tab transitions to the cart, which is **initially empty**. (The PROS Quote Content tab has three screens: Quote [the cart], Approval [Validation], and Proposal [Edition].)

The sales rep will browse the catalog to add services to the cart (see “Product Catalog” section below).

Within the cart, the sales rep **negotiates prices or discounts**; each **line item is scored** relative to service-specific discount ranges.

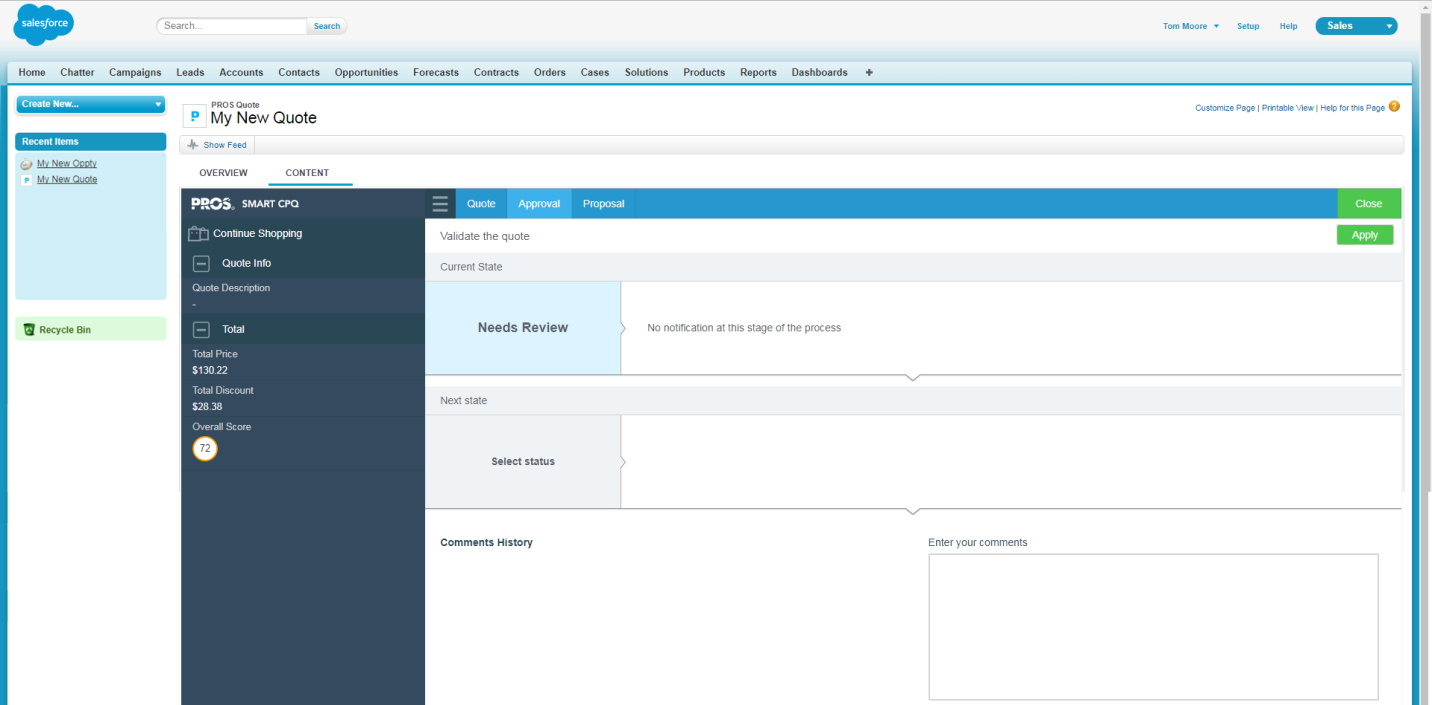


### Approval Process

#### Approval [Validation] Tab

When the quote is complete, the sales rep will click the Approval [Validation] tab and move the quote out of “Draft” status. The system will determine if the quote meets the requirements for internal approval. The system will make that assessment based on price of the service. If even one service is below the hard floor price, the quote will not be eligible for the next step in the quoting process (document generation). Sales rep will have to go back and adjust price until all approval rules are met. Once all rules are satisfied, the rep will have the ability to move the quote to the approved state. **Note: this does not mean the customer has accepted the quote. This is internal approval only.**

No approval workflow is required. Approval logic outlined in the “Approval Workflow” section.



#### Synchronize and Close

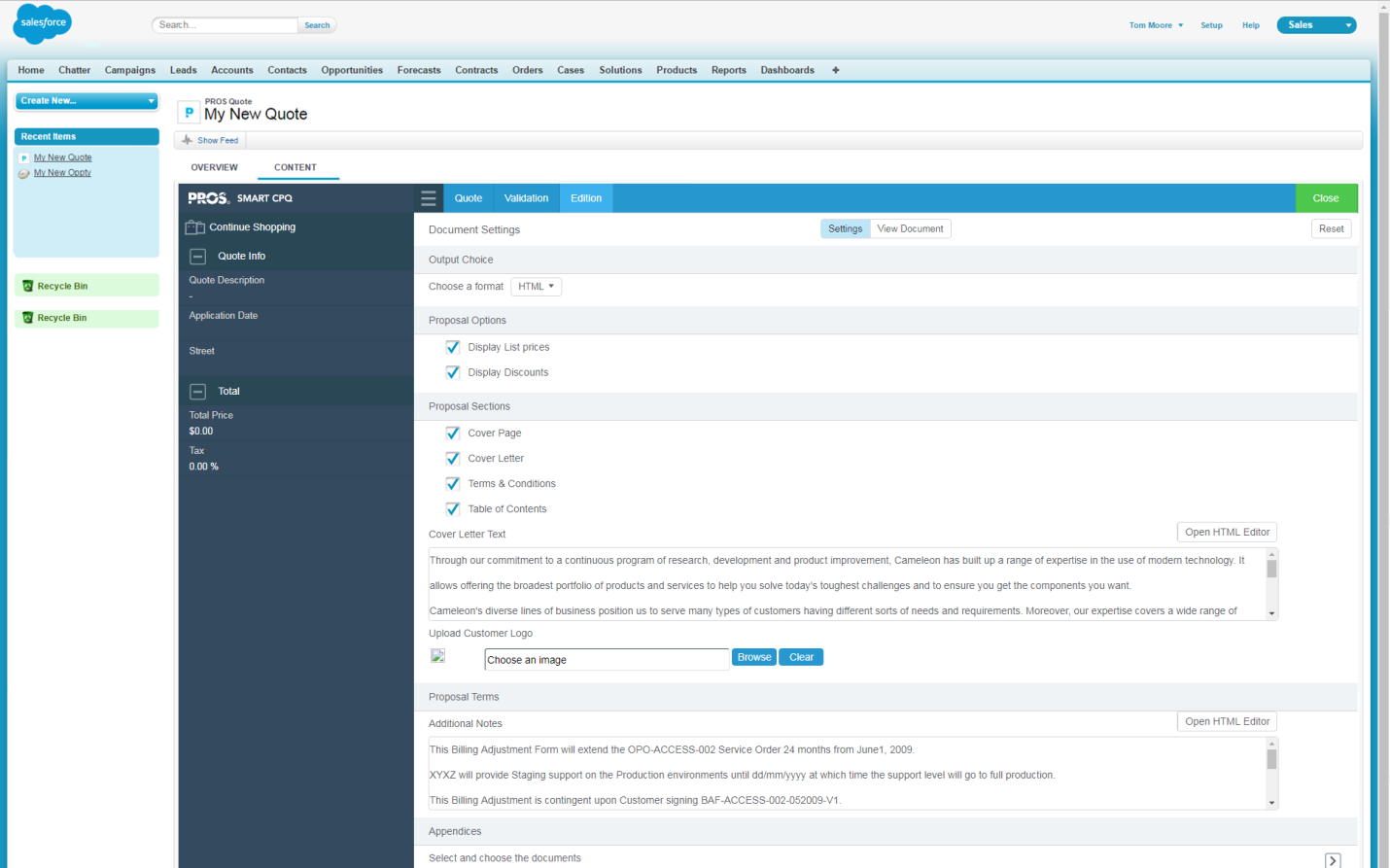
When the quote is complete, the sales rep will click “Close” to update Dynamics with the quote status and other quote details.

### Proposal Generation

The agreement document is illustrated in the “Proposal Generation” section below.

#### Proposal [Edition] Tab

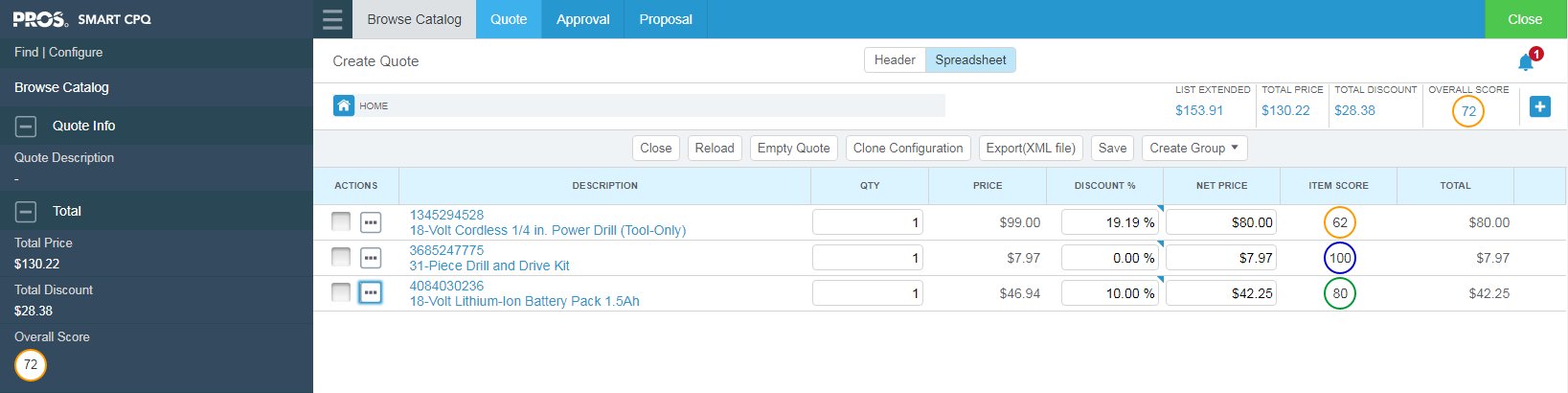
At any time the sales rep can generate a PDF agreement document via the Proposal [Edition] tab. The agreement document will be watermarked as “Draft” if the quote has not yet been approved. If the quote has been approved, the watermark will not exist.





# Cart Design

## Spreadsheet Columns



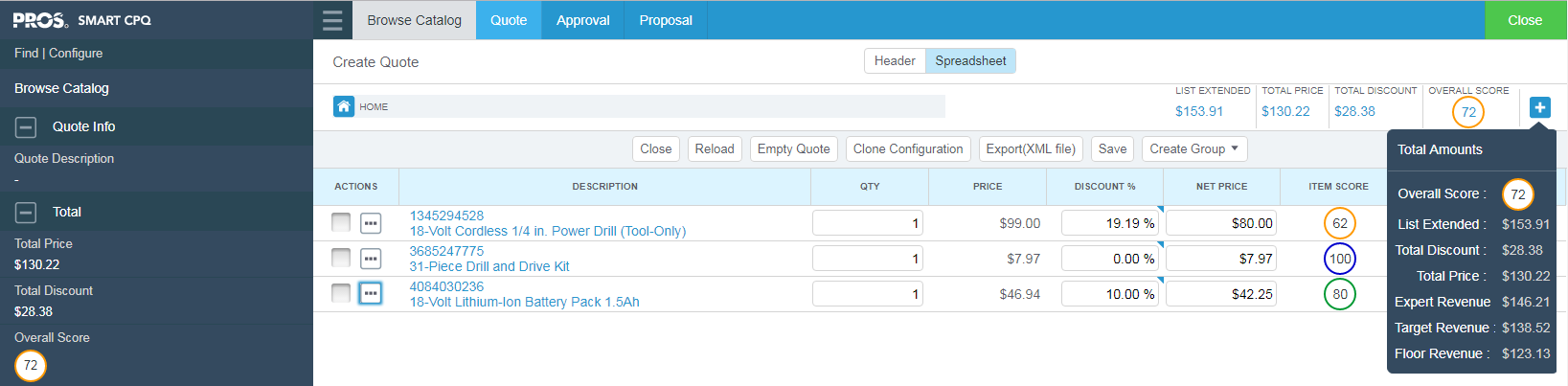
**Comments:**

**After the selection of items will this item score be displayed/shown to the users?**

**For Chubb, the following columns will be configured:**

* Description [of the service]
* Qty
* Price [catalog price]
* Discount (%)
* Discretionary Discount (%)
* Net Price [catalog price \* (1 - discount %)]
* Total [qty \* net price ]

## Top-Line and Left Sidebar Totals



**For Chubb, the following total fields will be displayed at the top:**

* List Extended [sum of (qty \* catalog price)]
* Total Price [sum of Total]
* Total Discount [list extended - total price]

**For Chubb, the following sidebar fields will be configured:**

Quote Info

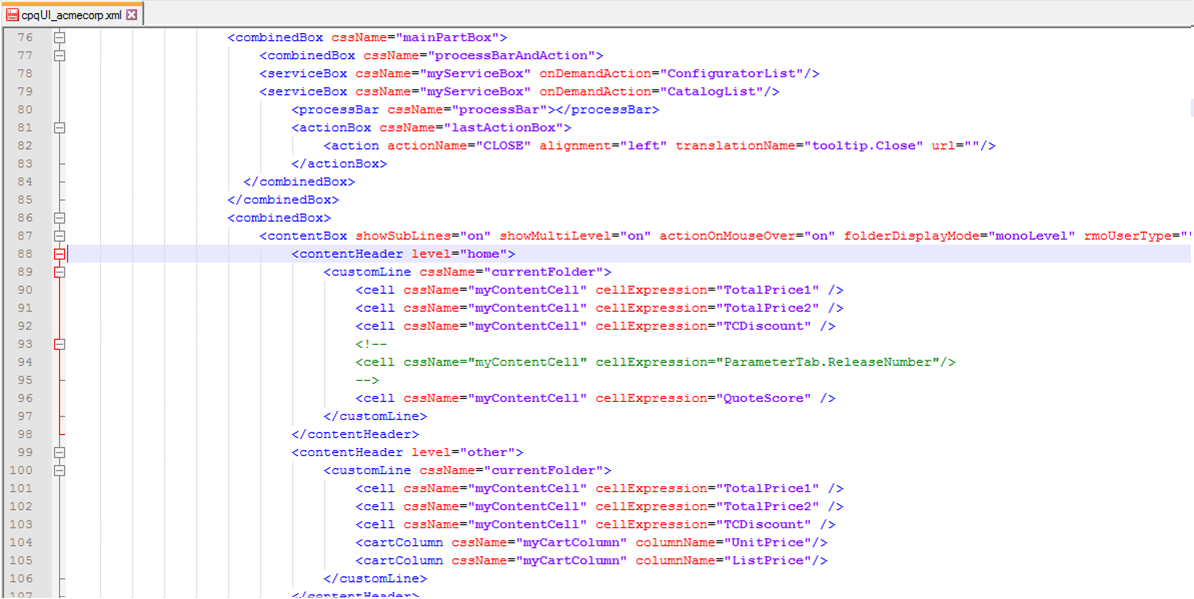
* Quote Description
* Quote Status

Total

* Total Price [sum of qty \* net price]
* Total Discount [list extended - total price]

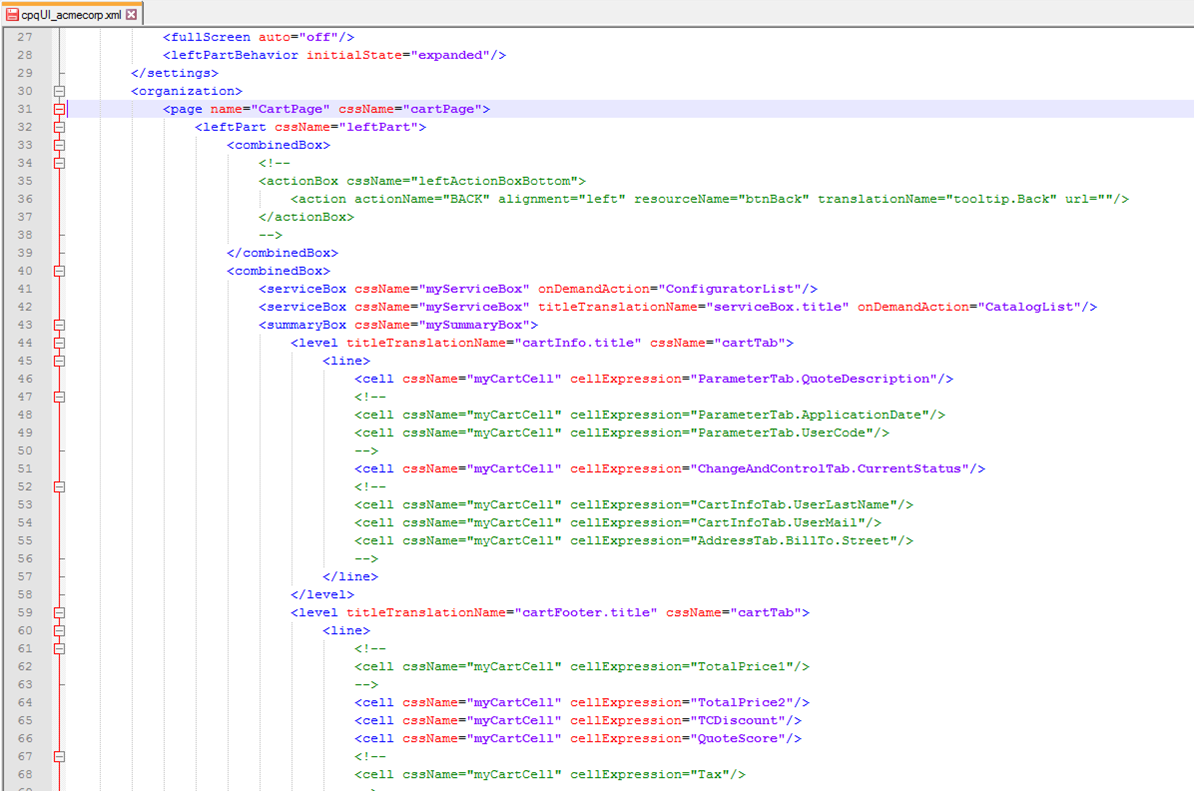
#### [For PROS] Configuration Notes for Top-Line Total Fields

The top-line total fields are configured outside the UI in the cpqUI\_<customer>.xml file (search for ‘contentHeader’):



#### [For PROS] Configuration Notes for Left Sidebar Fields

The sidebar fields are configured in the cpqUI\_<customer>.xml file (search for ‘cartTab’):



#### [For PROS] Configuration Notes for Left Sidebar Collapsed/Expanded

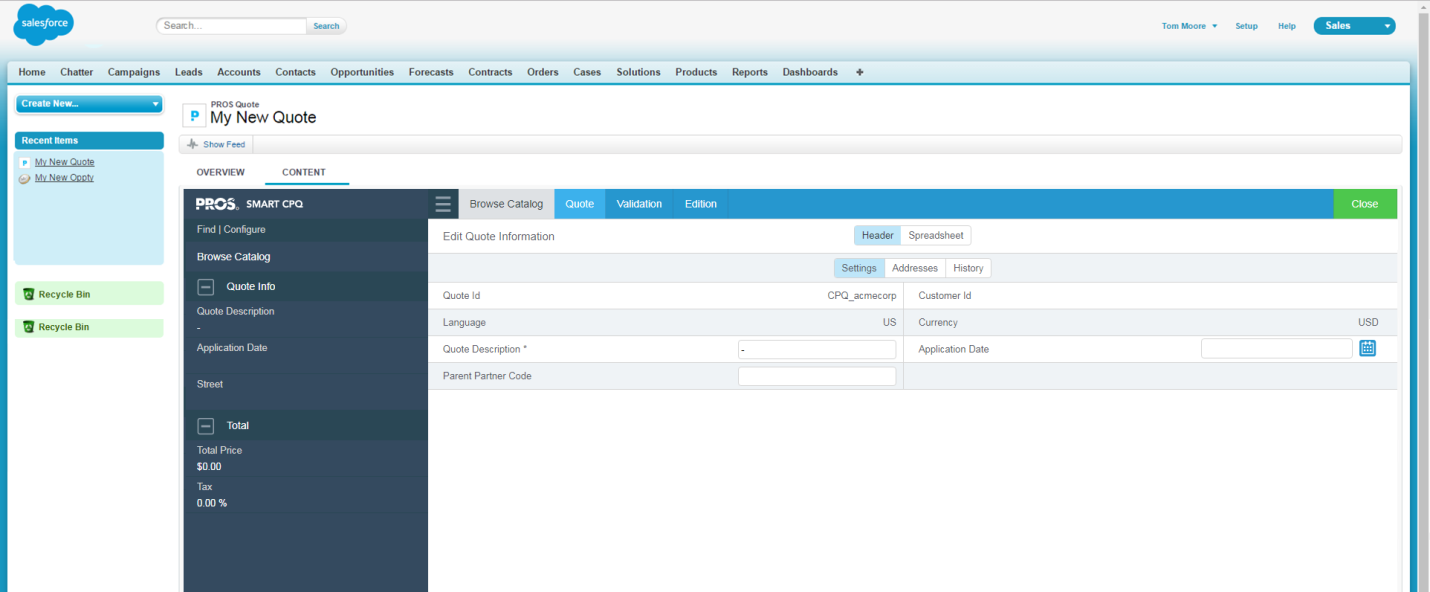
The sidebar can be configured to be initially collapsed or expanded [“collapsed” is typically preferred] in the cpqUI\_<customer>.xml file (search for ‘initialState’):



## Header Fields

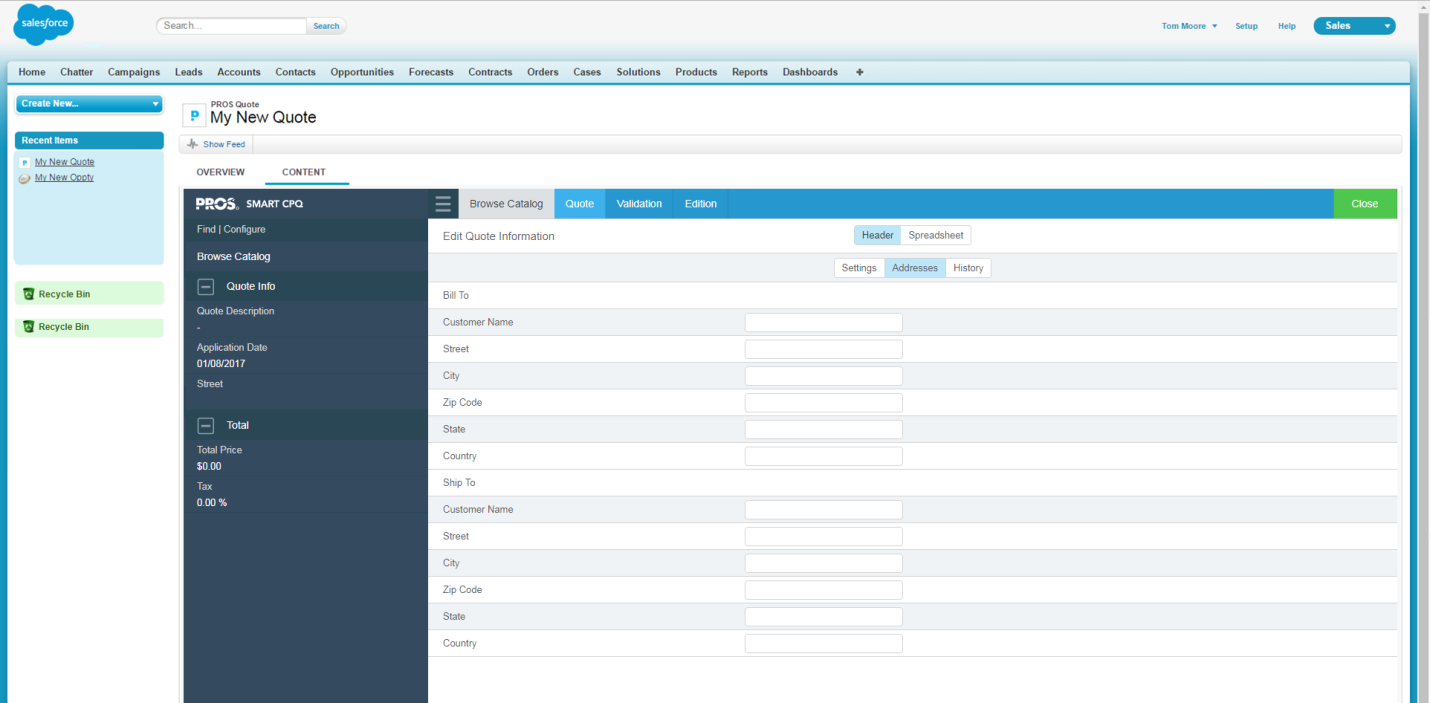
**For Chubb, the following Settings header fields (ParameterTab) minimally will be configured for reference by the Jasper proposal document:**

* ParameterTab.QuoteId
* ParameterTab.Currency



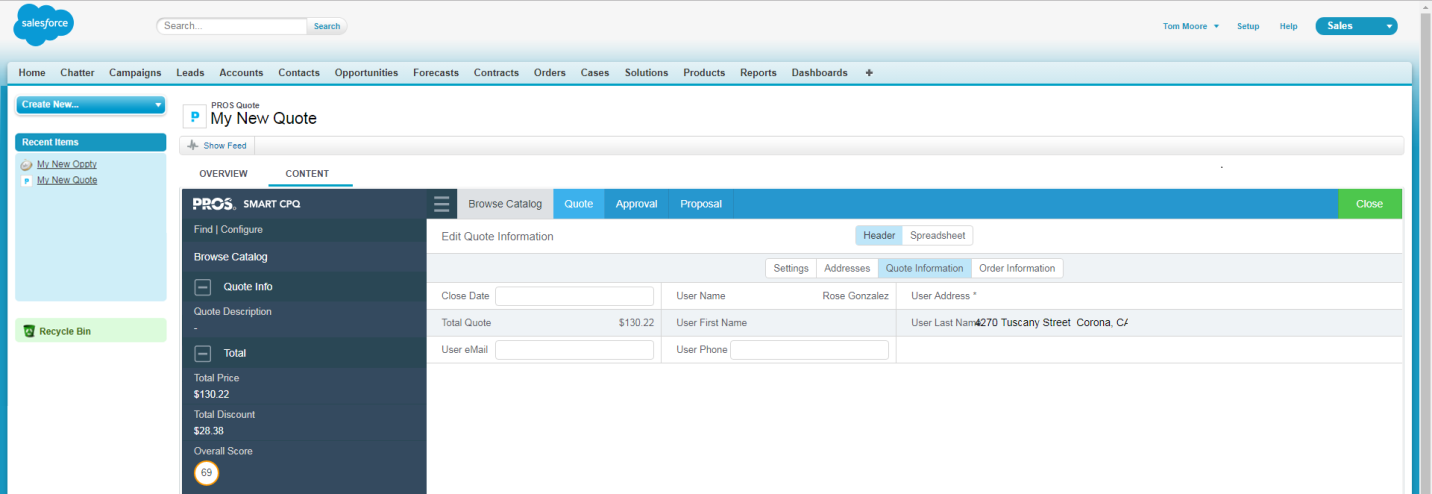
**For Chubb, the following Addresses header fields (AddressTab) minimally will be configured for reference by the proposal email:**

* AddressTab.SoldTo.eMail [email-to field]
* Lead.CompanyName
* Lead.BillingStreet
* Lead.BillingCity
* Lead.BillingPostalCode



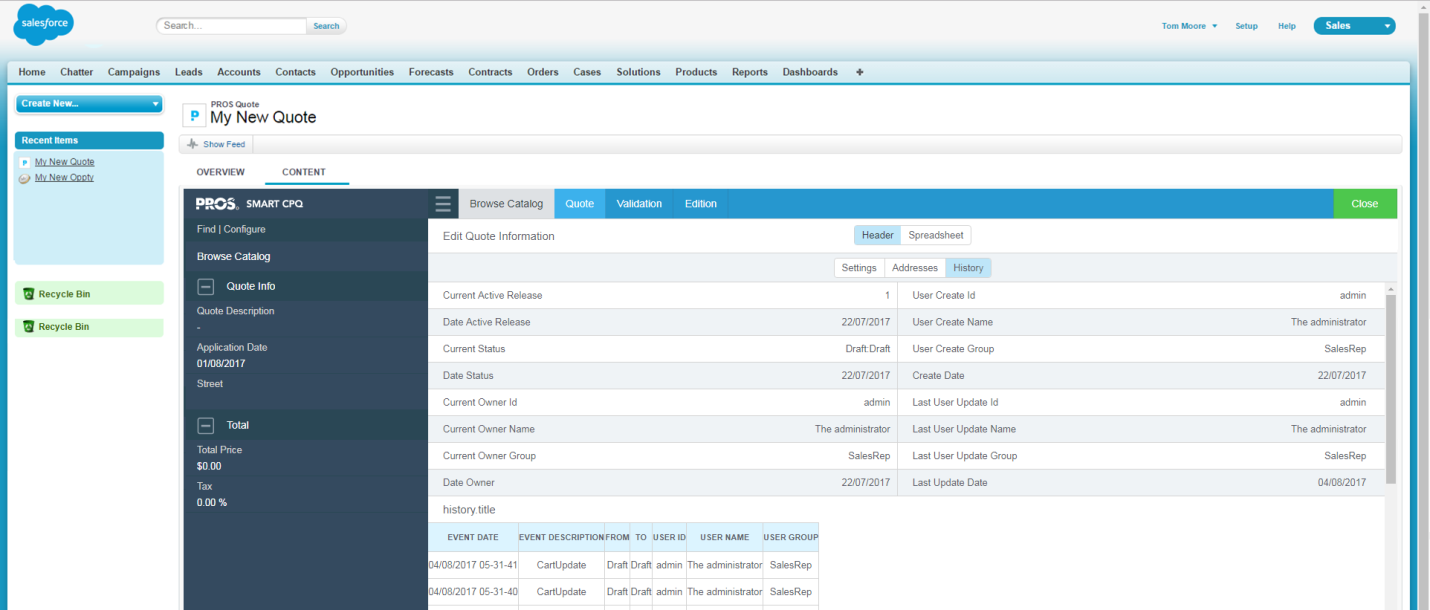
**For Chubb, the following Quote Information** **header fields** (**CartInfoTab) minimally will be configured for reference by the Jasper proposal document and by the proposal email:**

* CartInfoTab.CloseDate [proposal document]
* CartInfoTab.UserMail [email-from field]



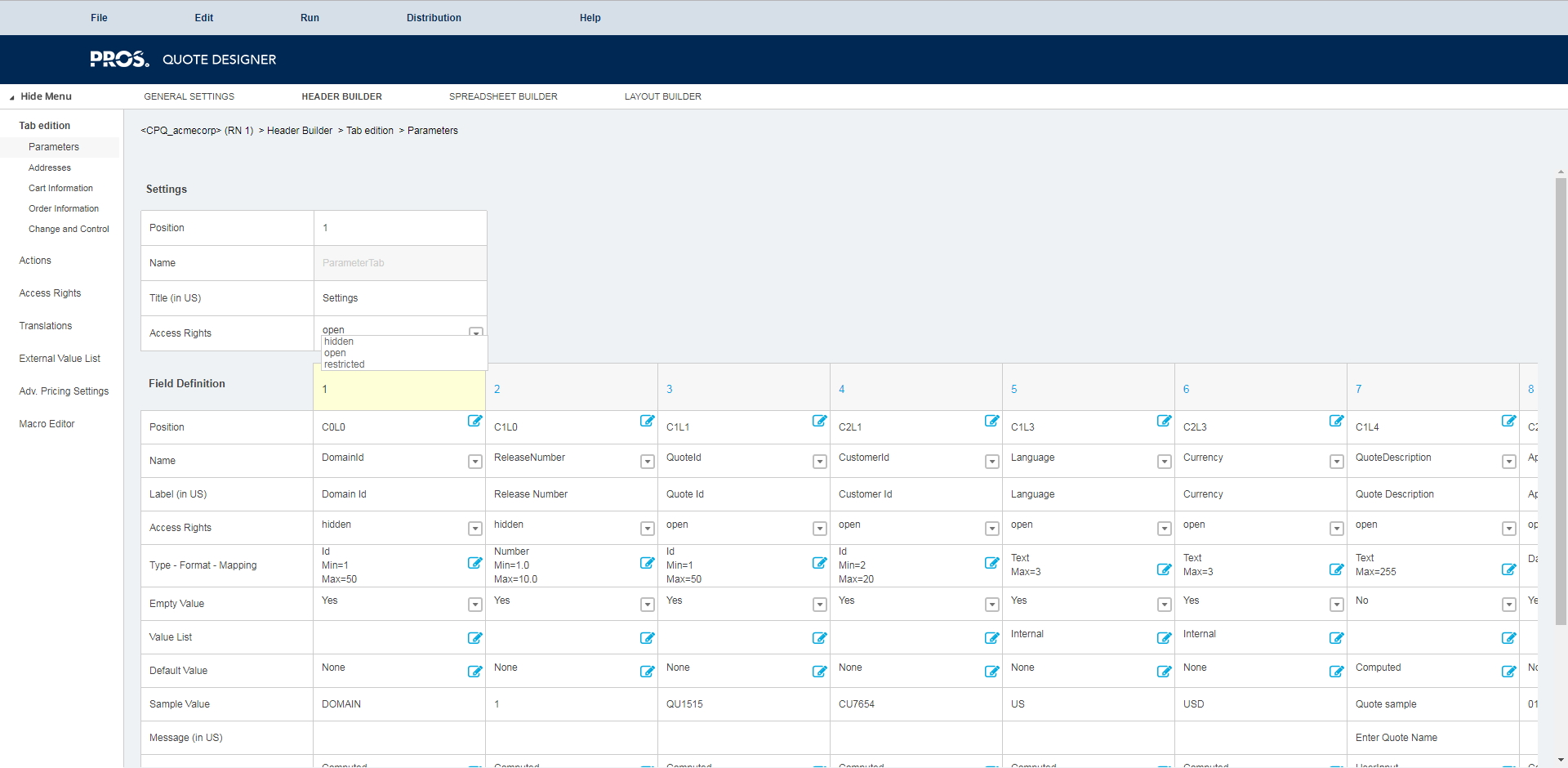
**For Chubb, the following History header fields (ChangeAndControlTab) minimally will be configured for reference by the Jasper proposal document:**

* ChangeAndControlTab.CreateDate

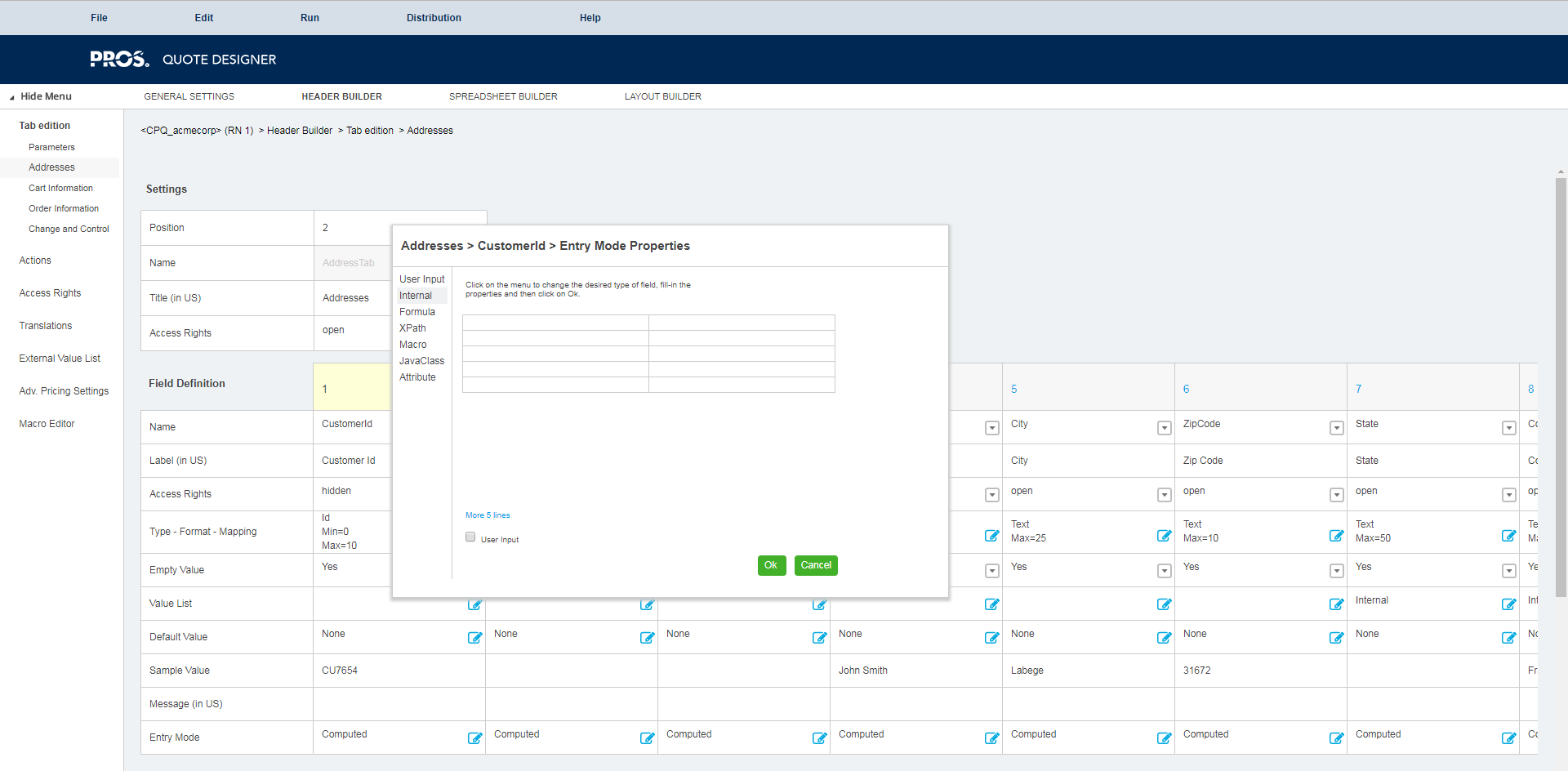


#### [For PROS] Configuration Notes for Header Tabs

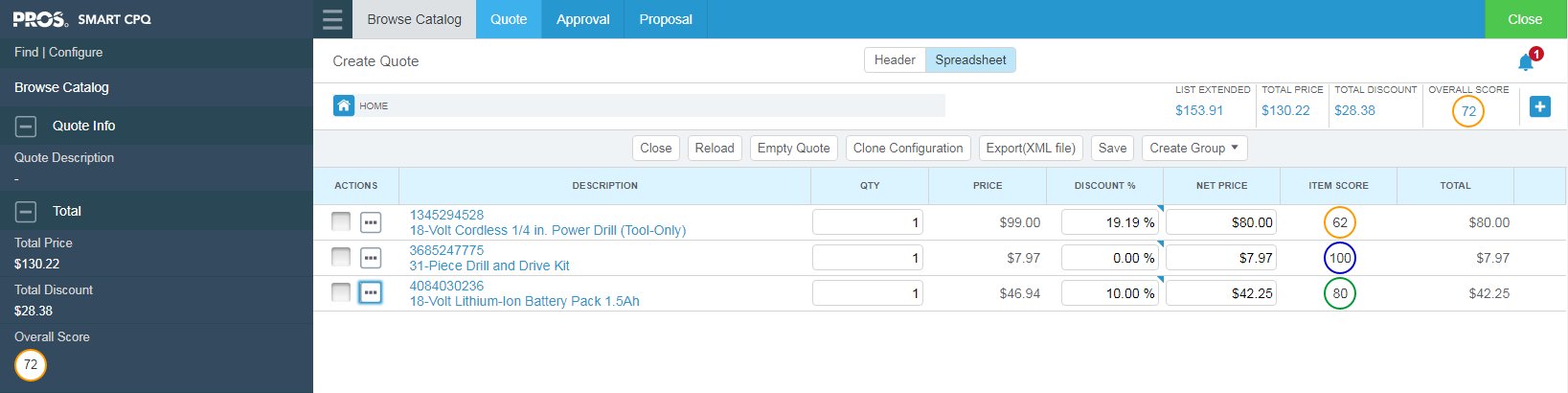
The header tabs can optionally be hidden by setting to ‘hidden’:



The header field on the Addresses tab have been configured as ‘Computed: Internal’ (which should automatically map to the Lead mapping object):



## OnDemand Grid Actions

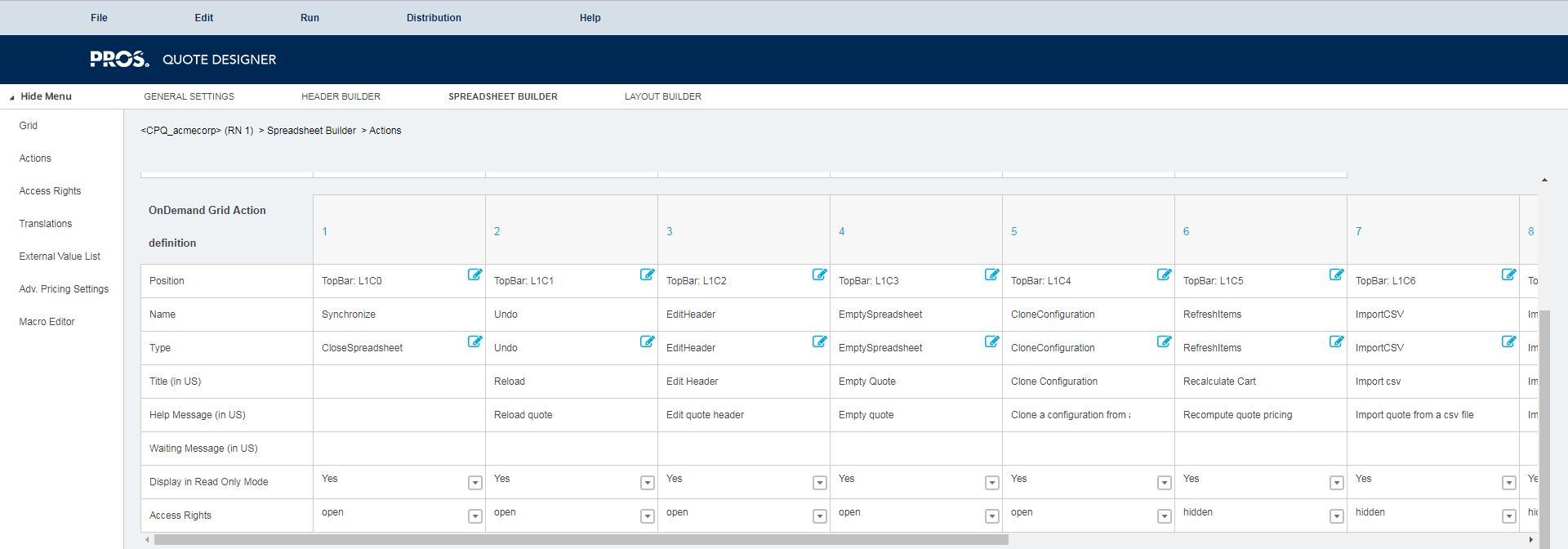


**For Chubb, the following grid actions will be configured:**

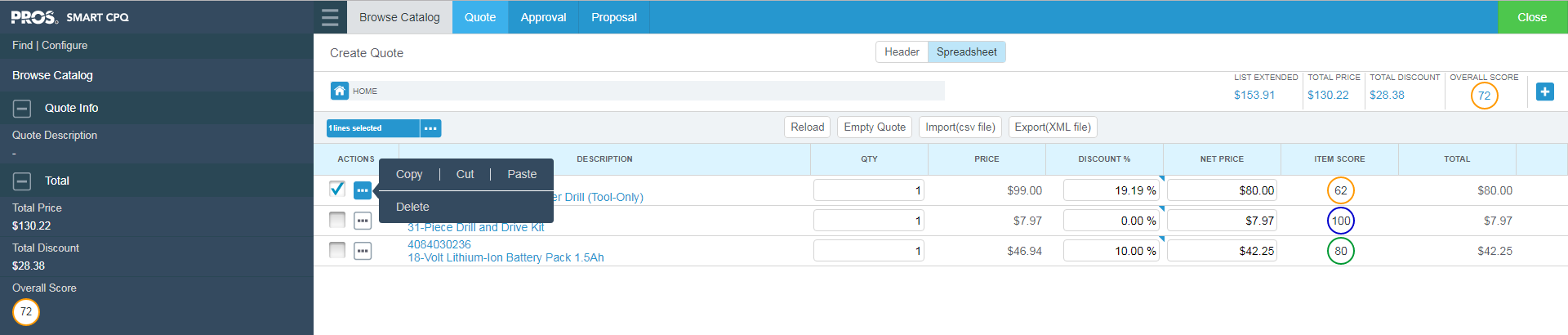
* Reload
* Empty Quote
* Clone Configuration

#### [For PROS] Configuration Notes for OnDemand Actions

Grid actions are enabled and disabled via Spreadsheet Builder/Actions/OnDemand Grid Actions within the cart model:

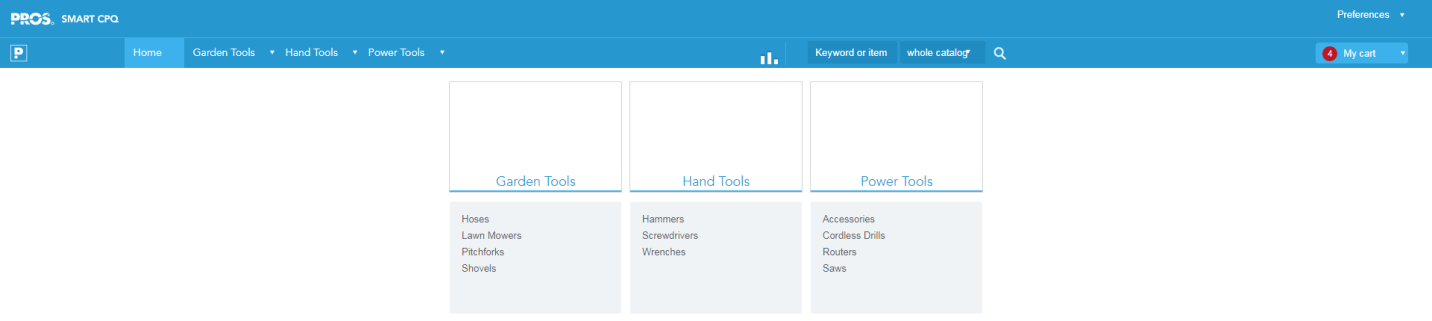


Menu and Line Actions refer to these (note: the title “Browse Catalog” is configured in CartServices.properties):



# Product Catalog

## Catalog Structure



**Comments:**

**As there are n’**

The catalog defines the organization of products within nested collections:

Acme Corporation (catAcme)

Garden Tools (col…)

Hoses (col…)

:

Hand Tools (col…)

Hammers (col…)

:

Power Tools (col….)

Accessories (col…)

:

The structure does NOT have to be strictly hierarchical. Products and collections can have multiple parents. For example, the entire product catalog could be organized under parallel hierarchies, e.g., the first hierarchy by industry and the second by product group or type:

**For Chubb, the catalog will initially only be used to select services. When equipment is within scope of implementation, a more sophisticated catalog structure will be required. The catalog structure for Chubb services is:**

Chubb Services (catServices)

One Time Assessment (col…)

Fire Risk Assessment

Fire Training

Multi-Year Agreements (col…)

PFX

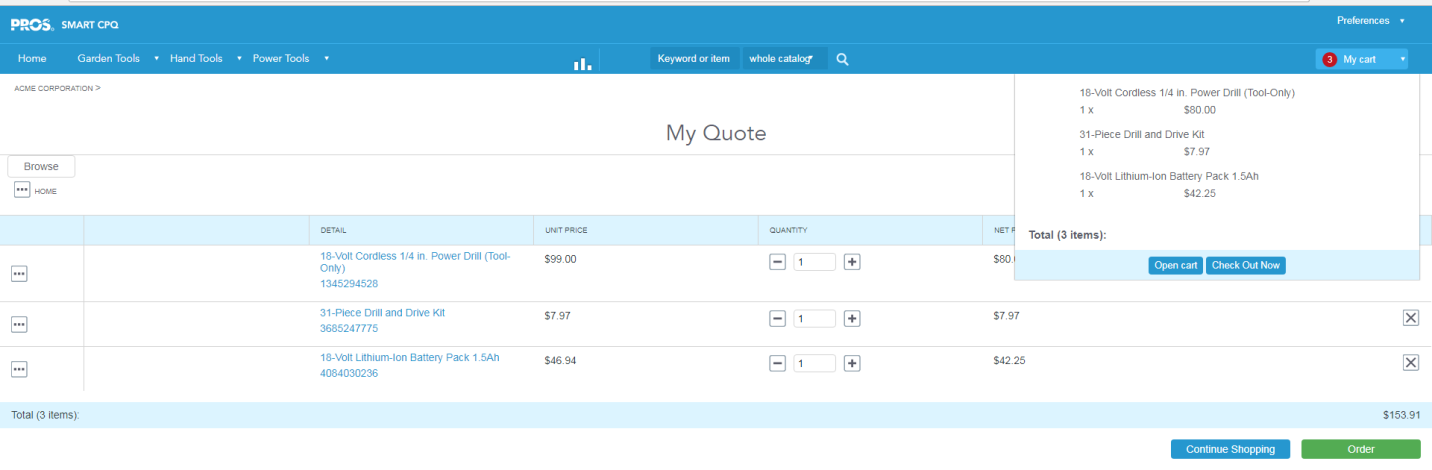
Security

Fire Systems



#### [For PROS] Configuration Notes for the “Cart Sticker”

Note: The “Open Cart” option in the “My Cart” dropdown which opens the “My Quote” screen has been disabled in the catalogUIDesktop file:



Note: The following line must be commented out in multiple places.

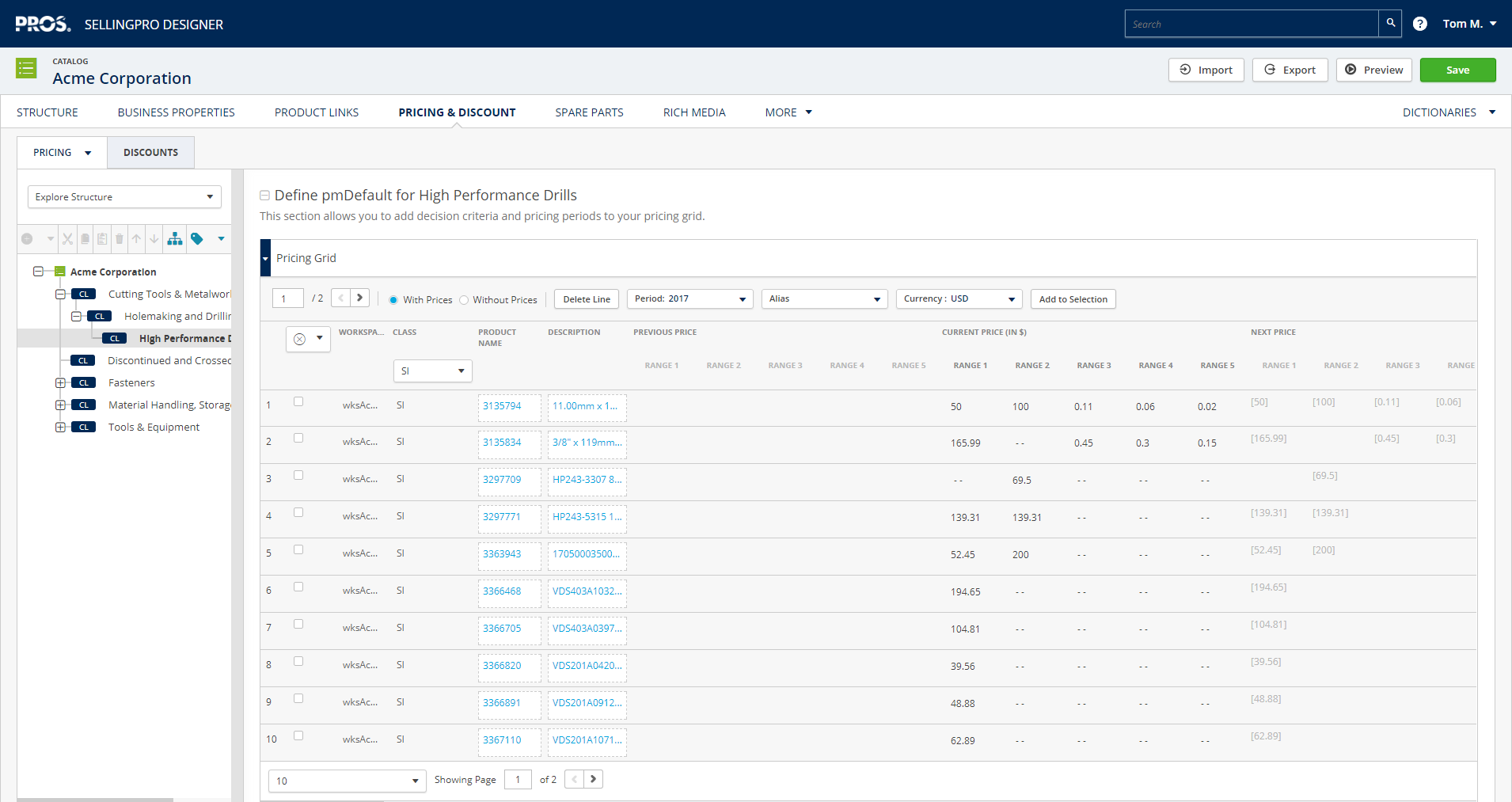


## Pricing and Discounts

Pricing Methods in CPQ will be configured to follow the following pricing logic by service. An additional column (range) will be created to allow global discounts to be applied by service. Catalog price will appear a 0 for the base configured services with adders dependent on the logic defined in the tables.



Product prices and discounts are manually entered into the catalog (see screenshot below).



## Business Properties

Business properties are effectively attributes of the items in the catalog. It can be an effective way for users to search for specific items based on attributes like color, size, etc.

**It is not expected for business properties to be useful to Chubb for the scope of this implementation. They will likely be added when quoting for equipment is implemented**

# Configuration Process

A configuration process is used to direct users through a list of questions or selections so that the appropriate service can be identified and price. It is often used when price is determined by a number of factors beyond the definition of the product (or in this case service) offered.

**For Chubb, we will utilize the structure that is currently deployed on the customer facing e-commerce solution.**

## Fire Risk Assessment

What is the size of the site? (radio button – single select)

* 1-2,000 sq. ft.
* 2,001-5,000 sq. ft
* 5,001-10,000 sq. ft
* 10,001-20,000 sq. ft
* 20,001-40,000 sq. ft.
* 40,001-80,000 sq. ft
* 80,001-120,000 sq. ft.
* 120,001-160,000 sq. ft

Do any of the following apply to the site? (check box – multi-select; option for rich media embedded: **Chubb to provide media**)

* Multi occupancy
* Stores or uses more than 50 L of highly flammable materials
* Provides sleeping accommodations
* Is a licensed premises

Multiple buildings on site?

* Yes/No (dropdown: default to ‘No’)

This configuration process is SEQUENTIAL and SINGLE LANE (i.e. next step in process is not dependent on selection in current step). There are dno constraints on choices. (i.e. selection of one entity does not restrict choices in next step; no constraints on selections based on customer type, etc.)

## Fire Training

What type of training is required? (Image select – single select; option for rich media embedded: **Chubb to provide media**)

* Basic
* Care Homes
* Commercial Kitchens
* Fire Warden
* Fire Extinguishers

How many people require training? (free text field)

This configuration process is SEQUENTIAL and SINGLE LANE (i.e. next step in process is not dependent on selection in current step). There are no constraints on choices. (i.e. selection of one entity does not restrict choices in next step; no constraints on selections based on customer type, etc.)

## PFX

Paid Service or Rental? (Image select – single select; option for rich media embedded: **Chubb to provide media**)

* Paid Service
  + Number of extinguishers on site (free text field)
  + Large spares and refills included?
    - Yes/No (dropdown: default to ‘Yes’)
* Rental
  + Number of extinguishers on site (free text field)
  + Contract Length (radio button – single select)
    - 3 Years
    - 5 Years

This configuration process is SEQUENTIAL and DUAL LANE. Choices available for 2nd step are dependent on selection in first step.

## Security

What is the building size? (radio button – single selection)

* Under 1,000 sq. ft.
* 1,000 – 2,000 sq. ft.
* 2,001 – 5,000 sq. ft.
* 5,001 – 10,000 sq. ft.
* 10,001 – 20,000 sq. ft.
* Over 20,000 sq. ft.

What is the business use/building type? (Image select – single select; option for rich media embedded: **Chubb to provide media**)

* Residential
* Retail
* Office
* Factory/Warehouse
* Education
* Other

When was system installed?

* Within 5 years
* 5 to 10 years
* More than 10 years

Is system monitored?

* Yes/No (dropdown: default to ‘No’)

Contract Type?

* Standard/Comprehensive (dropdown: default to ‘Standard’)

CONSTRAINT: If system is more than 10 years, Comprehensive can not be selected

Monitoring Required?

* Yes/No (dropdown: default to ‘No’)

Length of contract in years?

* 1/2/3/4/5 (dropdown: default to ‘1’)

This configuration process is SEQUENTIAL and SINGLE LANE (i.e. next step in process is not dependent on selection in current step).

## Fire Systems

**TBD**

# Approval Workflow

A quote is in “Draft” status while it is being constructed. When the quote is complete, the sale rep must indicate that by clicking the **Approval** [Validation]tab and selecting a next state.

The sales rep clicks “**Apply**” after selecting an available next state, and then clicks “**Close**” to synchronize the quote status and other quote details with Dynamics.

The approval rules dictate whether the sales rep will be able to either directly approve the quote. or must submit for approval.

**For Chubb, there will be no approval workflow. There will only be approval rules. This means that pricing is not reviewed as long as it doesn’t break approval rules**

Note: **A quote is locked once it moves out of draft status** and can only be edited again by moving it back into draft status.

## Approval Statuses

**For Chubb, the following approval statuses will be implemented:**

* Draft
* Accepted
* Denied

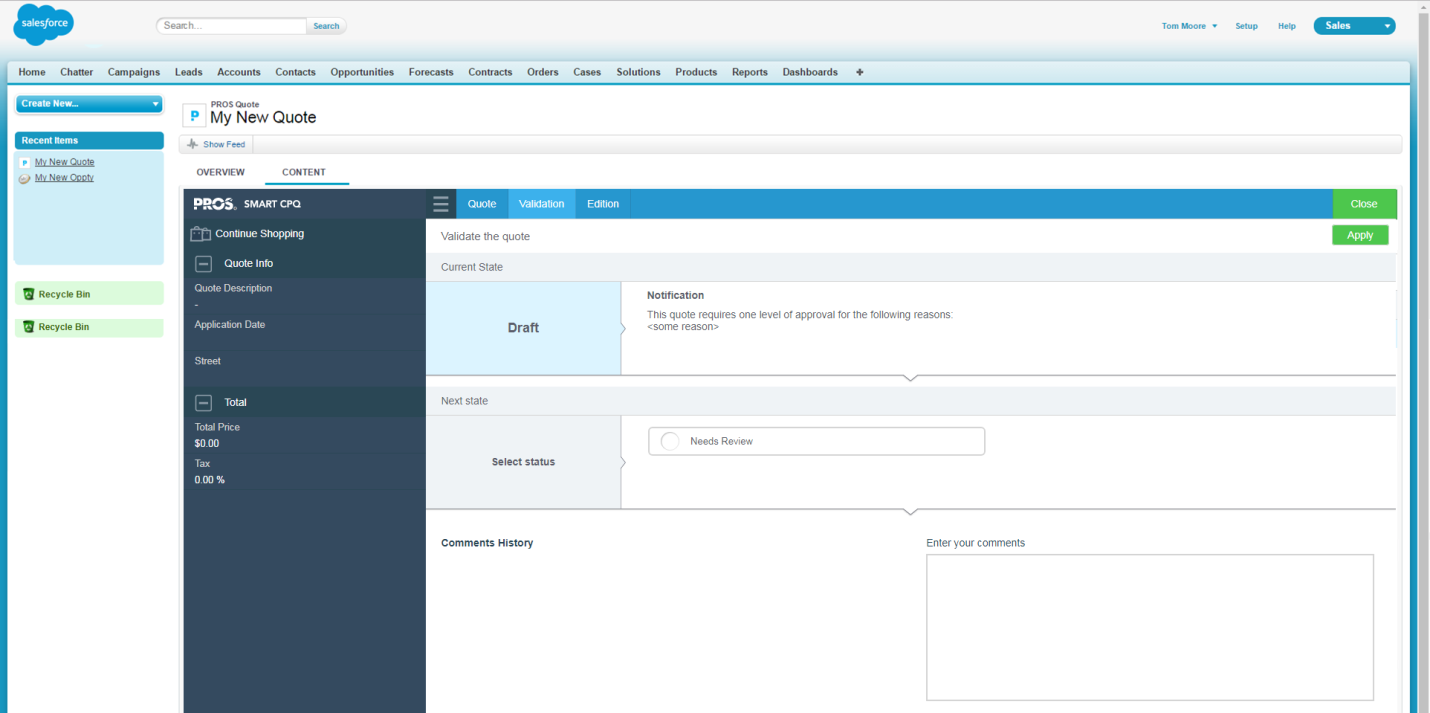
## Approval Rules

When the Approval [Validation] tab is clicked, Smart CPQ will determine available next-state options according to the following approval rules based on the **each line item’s total discount (global discount + discretionary discount)**:

* **Need total discount hard floors**

### Notification Panel

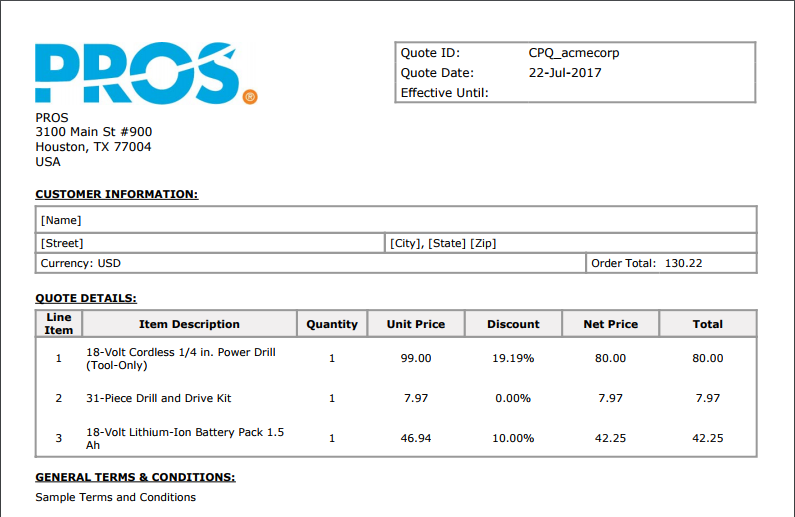
The **notification** panel, as illustrated in the following screenshot, **can be configured to indicate the reason pricing is denied.**



# Proposal Document

The CPQ solution will be configured to deliver one customer-facing agreement document in PDF format. It will contain a Chubb logo and standard information; header fields from the quote or lead; line items; totals from the quote; and terms & conditions or other standard boilerplate documentation.

NEED SAMPLE CHUBB DOCUMENTS



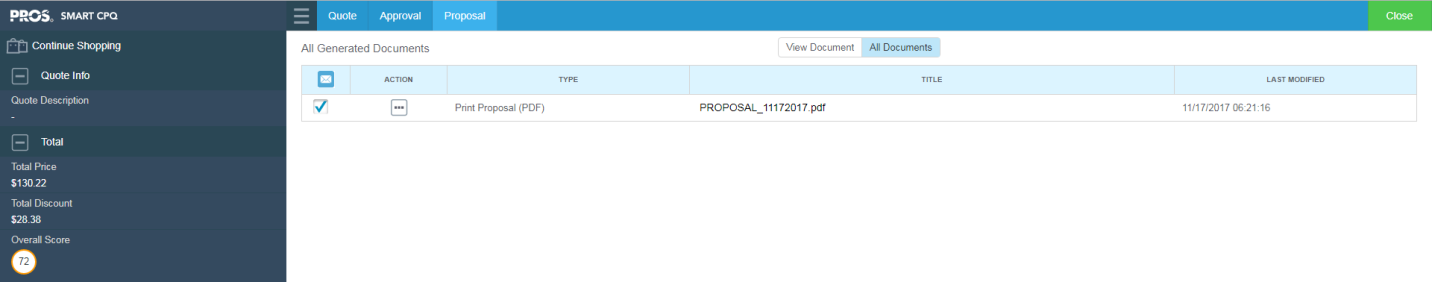
**For Chubb, the following columns will be configured on the proposal document:**

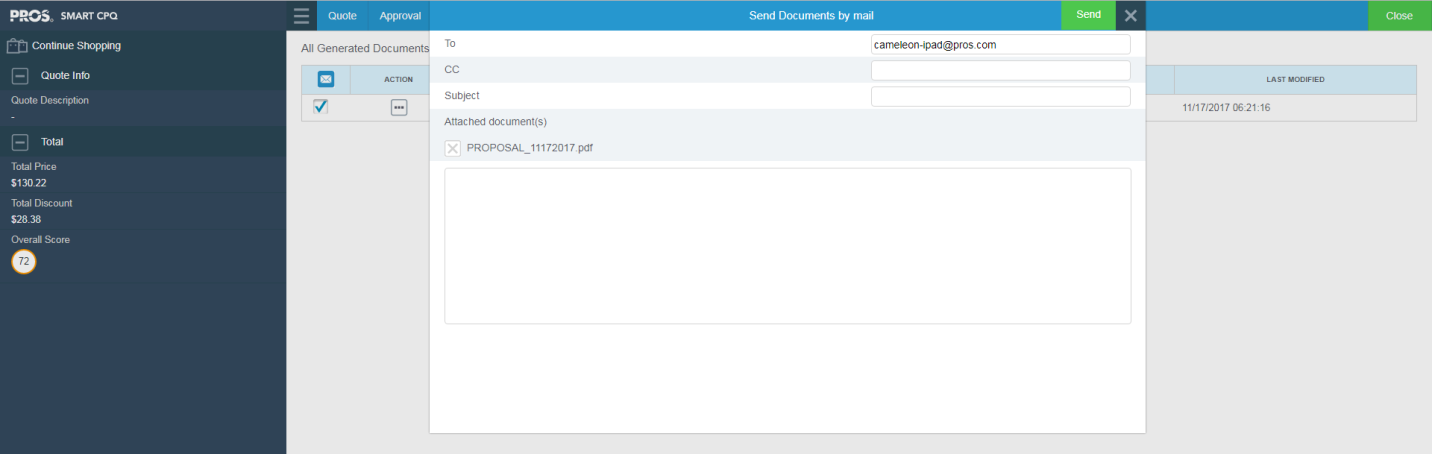
* Line Item
* Item Description
* Quantity
* Unit Price [Catalog Price]
* Discount [%]
* Net Price
* Total [Quantity \* Net Price ]

The following are to be provided by Chubb:

* Company logo
* Company address
* Terms and conditions language

### Proposal Email



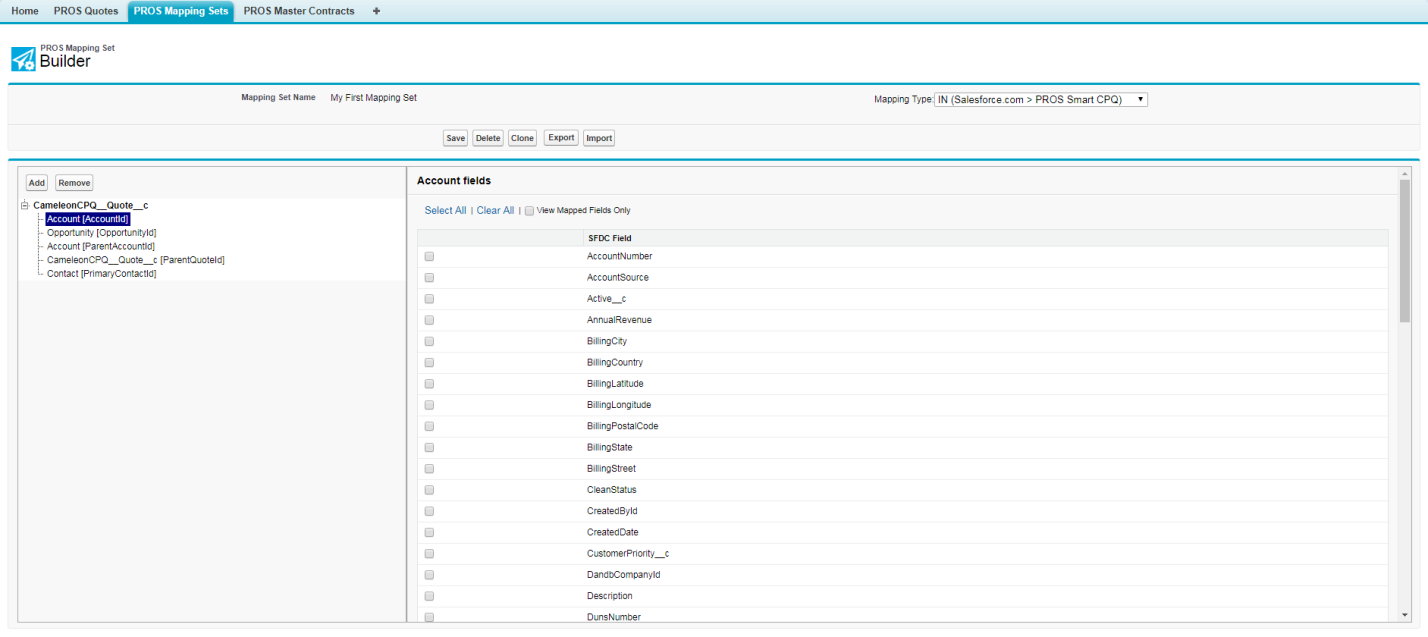


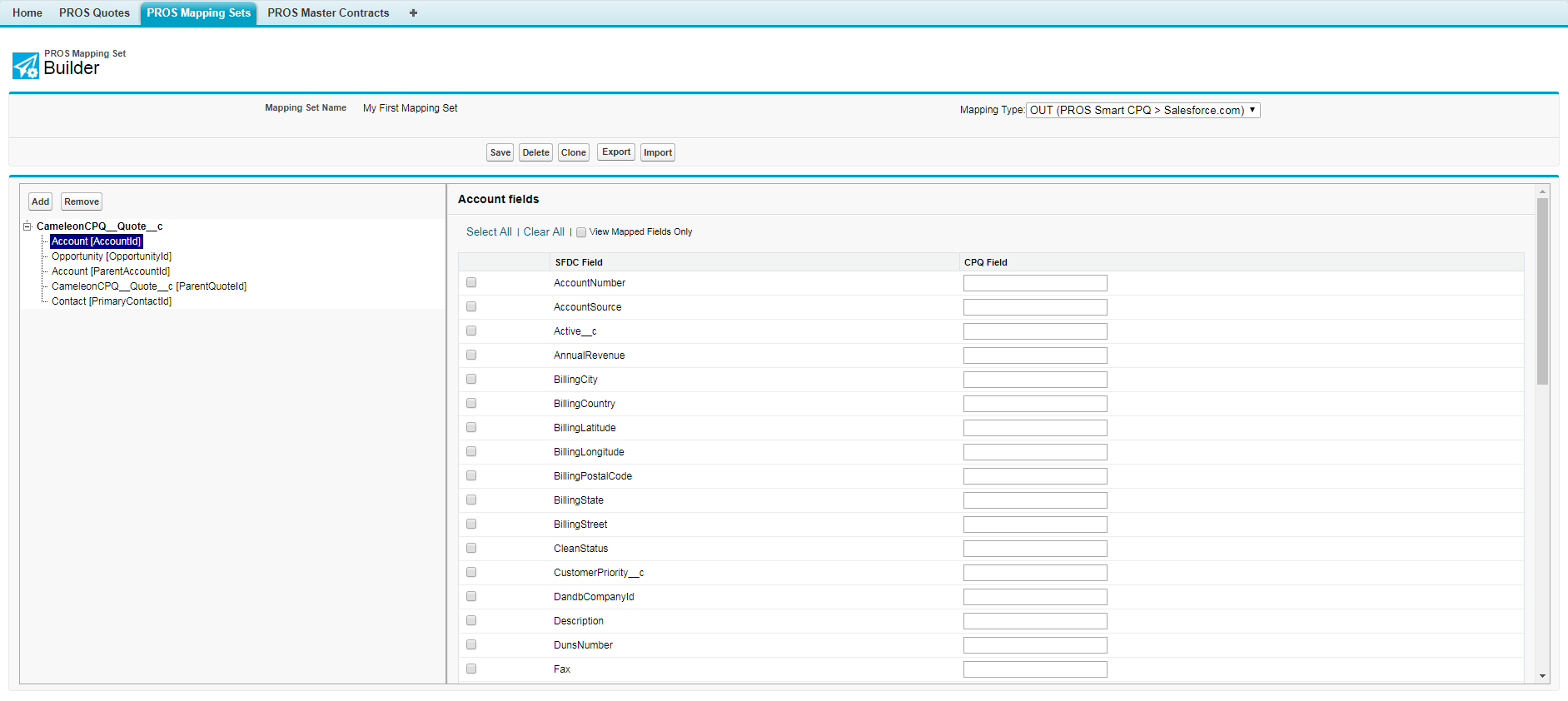
To enable email, Chubb needs to provide the following for setup in the **cameleon.properties** file:

mail.server=**CUSTOMER\_MAIL\_SERVER**  
# Host server port mail, only used in case of mail server authenticated  
#mail.port=**587**  
mail.protocol= **CUSTOMER\_MAIL\_PROTOCOL**  
mail.user= **CUSTOMER\_MAIL\_USER**  
mail.password= **CUSTOMER\_MAIL\_PWD**

# Map-In and Map-Out

Standard mapping in and mapping out between Smart CPQ and Dynamics is via PROS mapping sets with Dynamics:





# [For PROS] Quick Start Implementation Files

The following files are preconfigured with Quick Start:

* wksAcmeCorp.xlsx [content file (catalog and workspace)]
* catalog\_acmecorp.xml [settings file]
* configuration\_acmecorp.xml [settings file]
* createcart\_acmecorp.xml [settings file]
* catalogUIDesktop\_acmecorp.xml [theme/layout file]
* configurationUIDesktop\_acmecorp.xml [theme/layout file]
* cpqUI\_acmecorp.xml [theme/layout file]
* CPQ\_acmecorp.zip [cart model]
* Proposal.zip [Jasper proposal document]

The following file exists within the Quote Designer:

* cartServices.properties [catalog service and proposal service]
* cartSettings.properties [user group settings]

The following file exists on the server:

* cameleon.properties [email server for proposal emails]

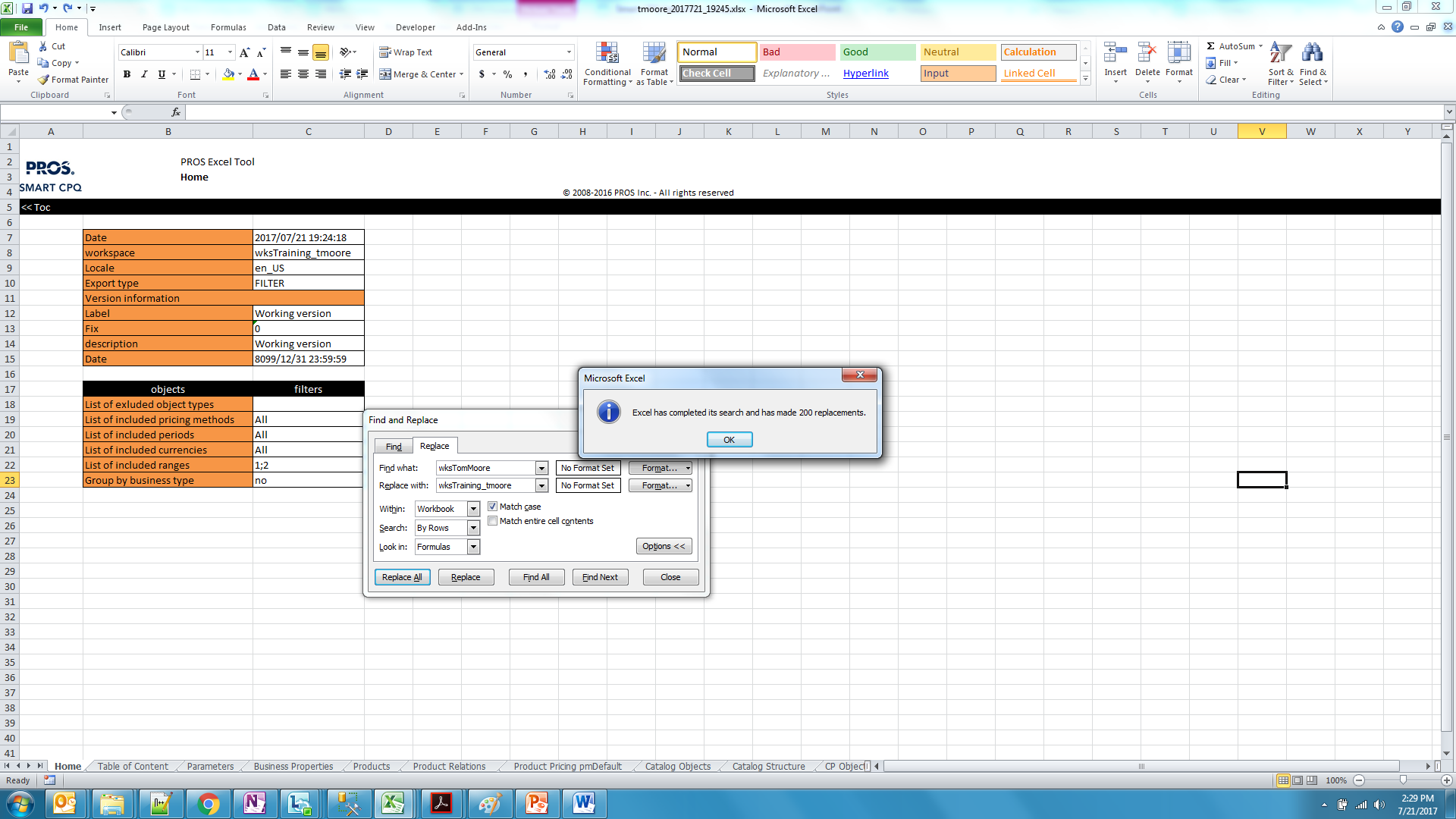
The following must be provided by Chubb:

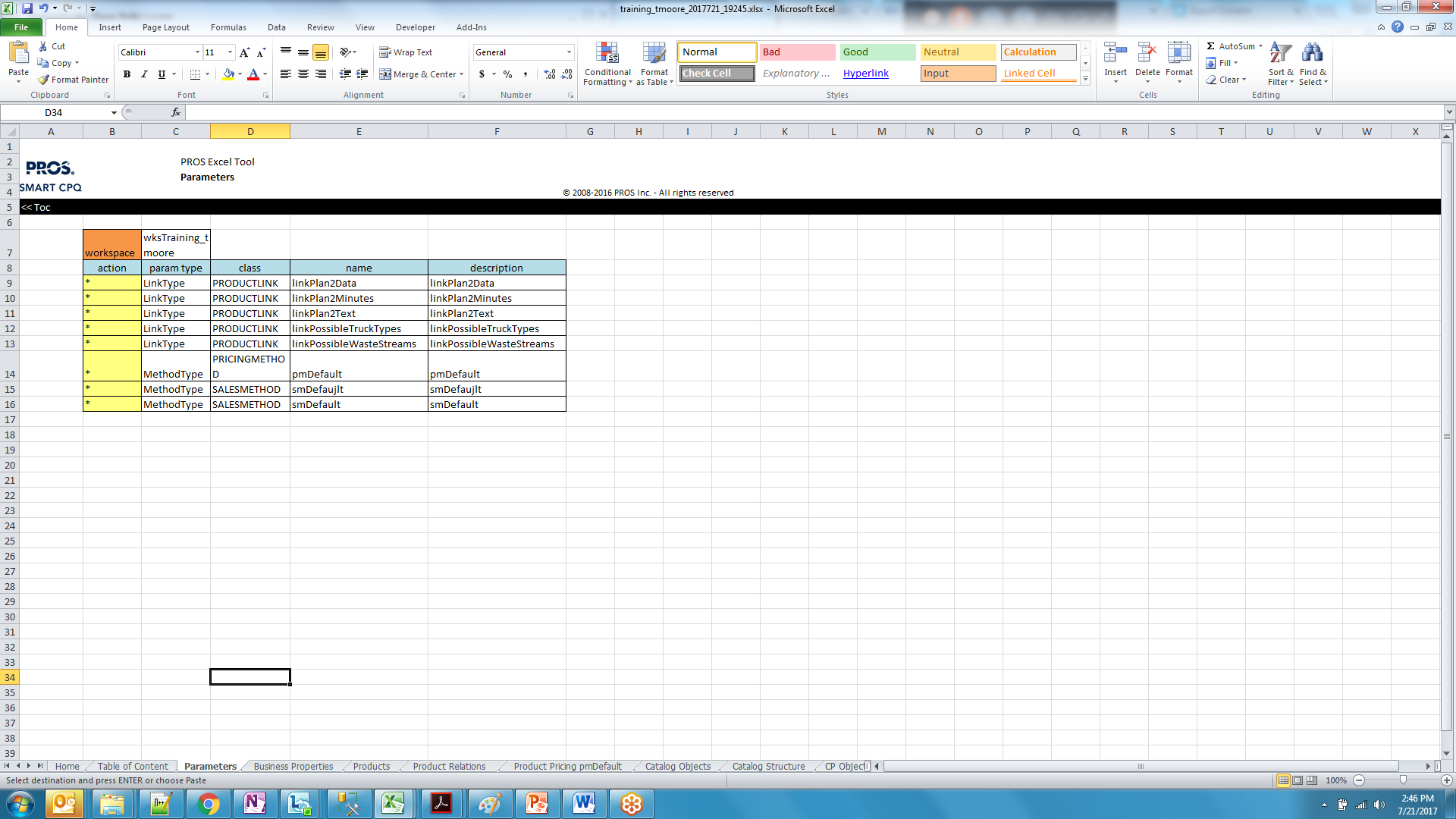
* Catalog logo (company logo in catalog)
* Company logo (for proposal document)
* Company address
* Terms and conditions language
* Email server (to send proposal emails)
* Email protocol
* Email login

## Excel Content File

**Rename** and **edit** the included Excel content file as follows, and then **import** into the CPQ environment:

* **wksAcmeCorp.xlsx**
  + Rename file to ‘wks<Customer>.xlsx’
  + Replace all occurrences of ‘wksAcmeCorp’ with ‘wks<Customer>’ (referenced in catalog\_<customer>.xml ~~and configuration\_<customer>.xml~~ settings file below)
  + Replace all occurrences of ‘catAcme’ with ‘cat<Customer>’ (referenced catalog\_<customer>.xml settings file below)
  + Replace all occurrences of ‘cpAcme’ with ‘cp<Customer>’ (referenced configuration\_<customer>.xml settings file below)
  + **Build out the “cat<Customer>” catalog structure, business properties, and prices and discounts**
    - Catalog Objects tab
    - Products tab
    - Catalog Structure tab
    - Product Pricing tab
  + Change any blank yellow fields to ‘\*’

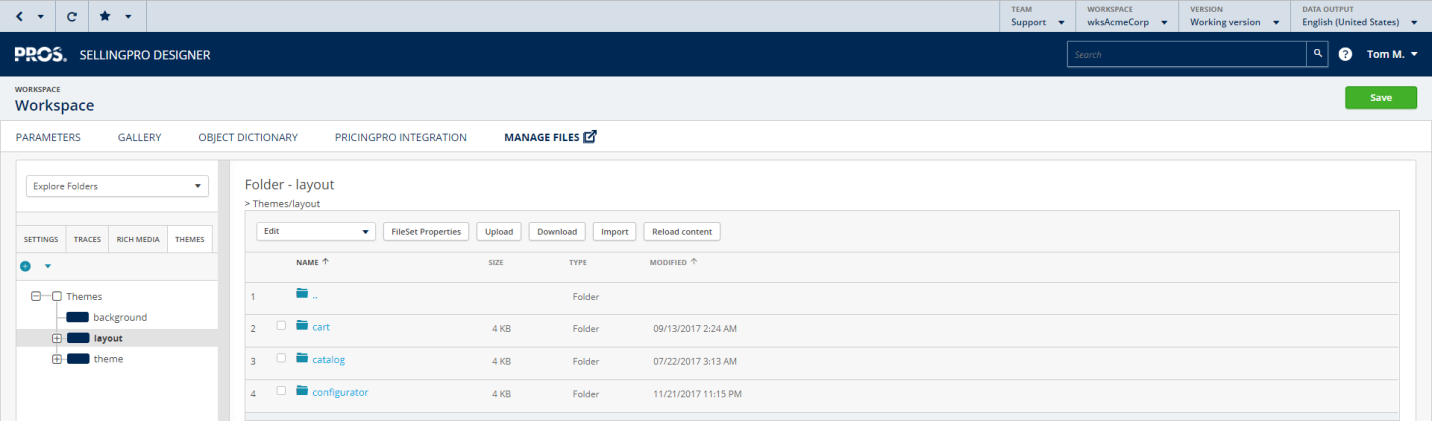




## Theme/Layout Files

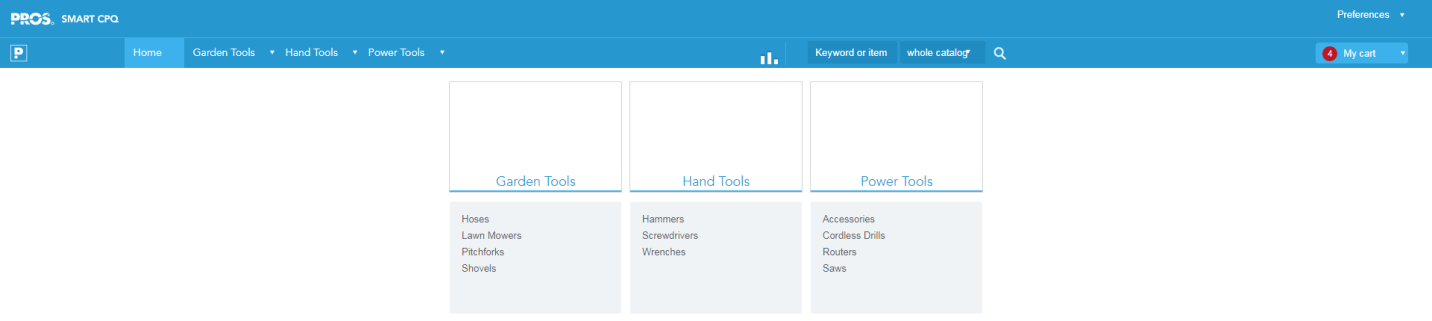
**Rename** each of the included theme/layout files as follows, and then **upload** into the CPQ environment:

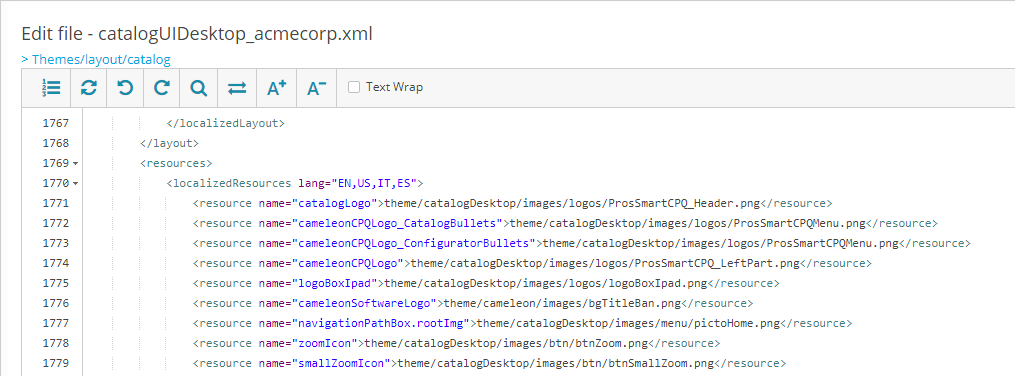
* **catalogUIDesktop\_acmecorp.xml**
  + Rename file to ‘catalogUIDesktop\_<customer>.xml’ (referenced in cartServices.properties below)
* **configurationUIDesktop\_acmecorp.xml**
  + Rename file to ‘configurationUIDesktop\_<customer>.xml’ (referenced in cartServices.properties below)
* **cpqUI\_acmecorp.xml**
  + Rename file to ‘cpqUI\_<customer>.xml’ (referenced in createcart\_<customer>.xml settings file below)



## Catalog Logo

The PROS logo on the catalog screen can be replaced with a  **customer logo**:



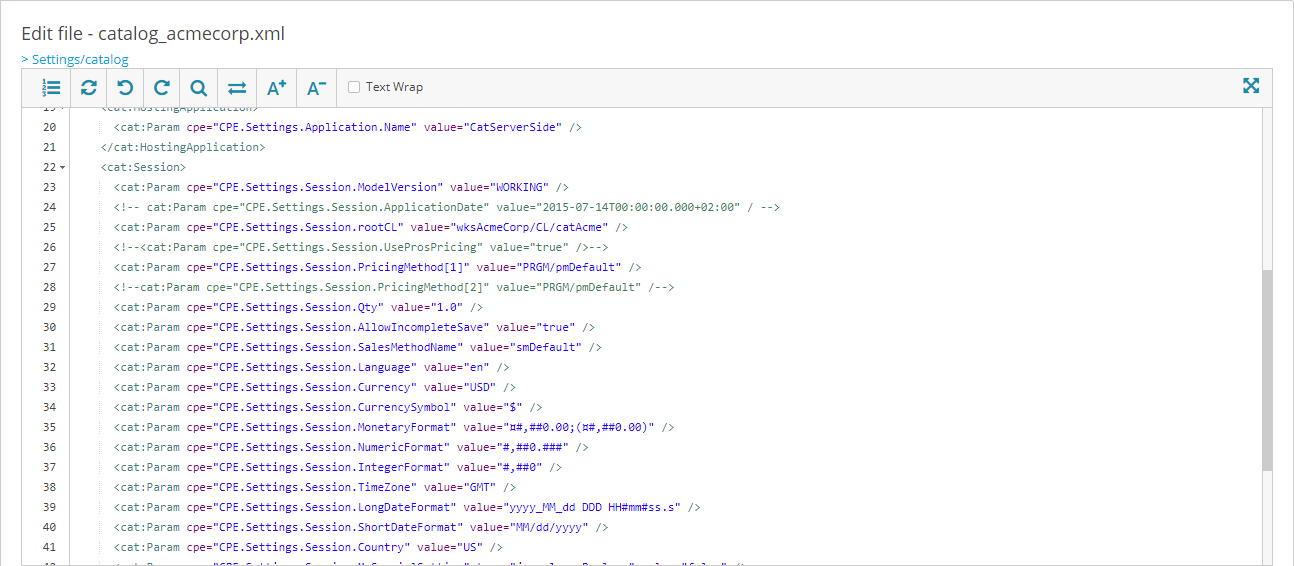


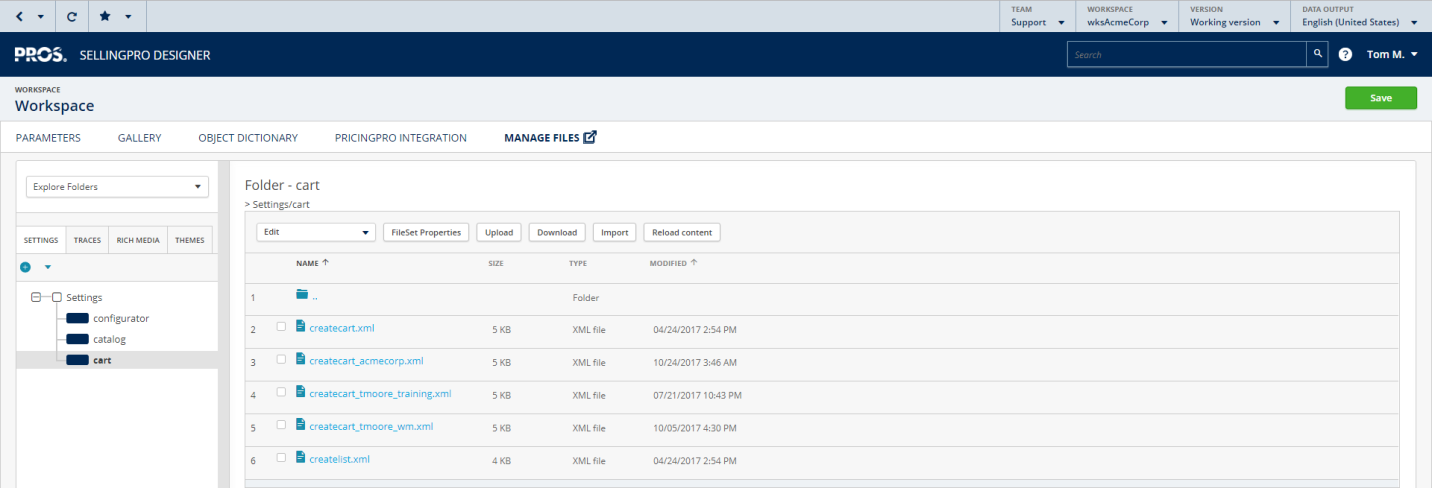


## Settings Files

**Rename** and **edit** each of the included settings files as follows, and then **upload** into the CPQ environment:

* **catalog\_acmecorp.xml** (identifies root catalog [cat<Customer>] and the pricing method for all catalog items [PRGM/pmDefault])
  + Rename file to ‘catalog\_<customer>.xml’ (referenced in cartServices.properties below)
  + Replace ‘wksAcmeCorp/CL/catAcme’ with ‘wks<Customer>/CL/cat<Customer>’ (defined in wks<Customer>.xlsx above)
* **configuration\_acmecorp.xml**
  + Rename file to ‘configuration\_<customer>.xml’ (referenced in cartServices.properties below)
  + Replace ‘wksAcmeCorp’ with ‘wks<Customer>’ (defined in wks<Customer>.xlsx above)
  + Replace ‘cpAcme’ with ‘cp<Customer>’ (defined in wks<Customer>.xlsx above)
* **createcart\_acmecorp.xml**
  + Rename file to ‘createcart\_<customer>.xml’ (run file specified in preview)
  + Replace ‘CPQ\_acmecorp’ with ‘CPQ\_<customer>’ (cart model .zip below)
  + Replace ‘ cpqUI\_acmecorp’ with ‘cpqUI\_<customer>’ (theme/layout file above)





## cartServices.properties

Modify the cartServices.properties file within the Quote Designer as follows:

* Download **cartServices.properties** [within Quote Designer, under “Distribution / **Services & Default Data** / Download”]
* Add CAT<customer> services (references cat theme/layout and settings files above):

### Quick Start ###

service.**CATacmecorp**.catalogType=CT7

service.**CATacmecorp**.xmlLayoutURI=**catalogUIDesktop\_acmecorp.xml**

service.**CATacmecorp**.xmlCatalogURI=**catalog\_acmecorp.xml**

service.**CATacmecorp**.xmlSavedCatalogURI=saved-catalog-EXT.xml

#service.**CATacmecorp**.FR=Mon Catalogue

service.**CATacmecorp**.US=Browse Catalog

* Add CONF<customer> services (references conf theme/layout and settings files above):

### Quick Start ###

service.**CONFacmecorp**.confType=CP7

service.**CONFacmecorp**.xmlLayoutURI=**configuratorUIDesktop\_acmecorp.xml**

service.**CONFacmecorp**.xmlConfigurationURI=**configuration\_acmecorp.xml**

service.**CONFacmecorp**.xmlSavedConfigurationURI=saved-configuration\_Trainingservice.xml

#service.**CONFacmecorp**.FR=Configurateur

service.**CONFacmecorp**.US=Configure Something

* Add PROPOSAL services if missing (references Jasper proposal document):

service.PROPOSAL.appBuiltXML=<origin type="CART" used="1" descr="Cart (Current)"><form used="1" descr="Form"></form></origin>

service.PROPOSAL.appBuiltXMLPolicy=full

service.PROPOSAL.appBuiltXMLType=complete

service.PROPOSAL.encoding=UTF-8

service.PROPOSAL.mainDataSourceType=appBuiltXML

#service.PROPOSAL.outputFormat=UserChoice

**service.PROPOSAL.outputFormat=PDF**

service.PROPOSAL.reportFile=**MainReport**.jrxml

service.PROPOSAL.reportLanguage=Entity

#service.PROPOSAL.reportMultiLanguage=Yes

**service.PROPOSAL.reportMultiLanguage=No**

service.PROPOSAL.xpathXML=/**origin**

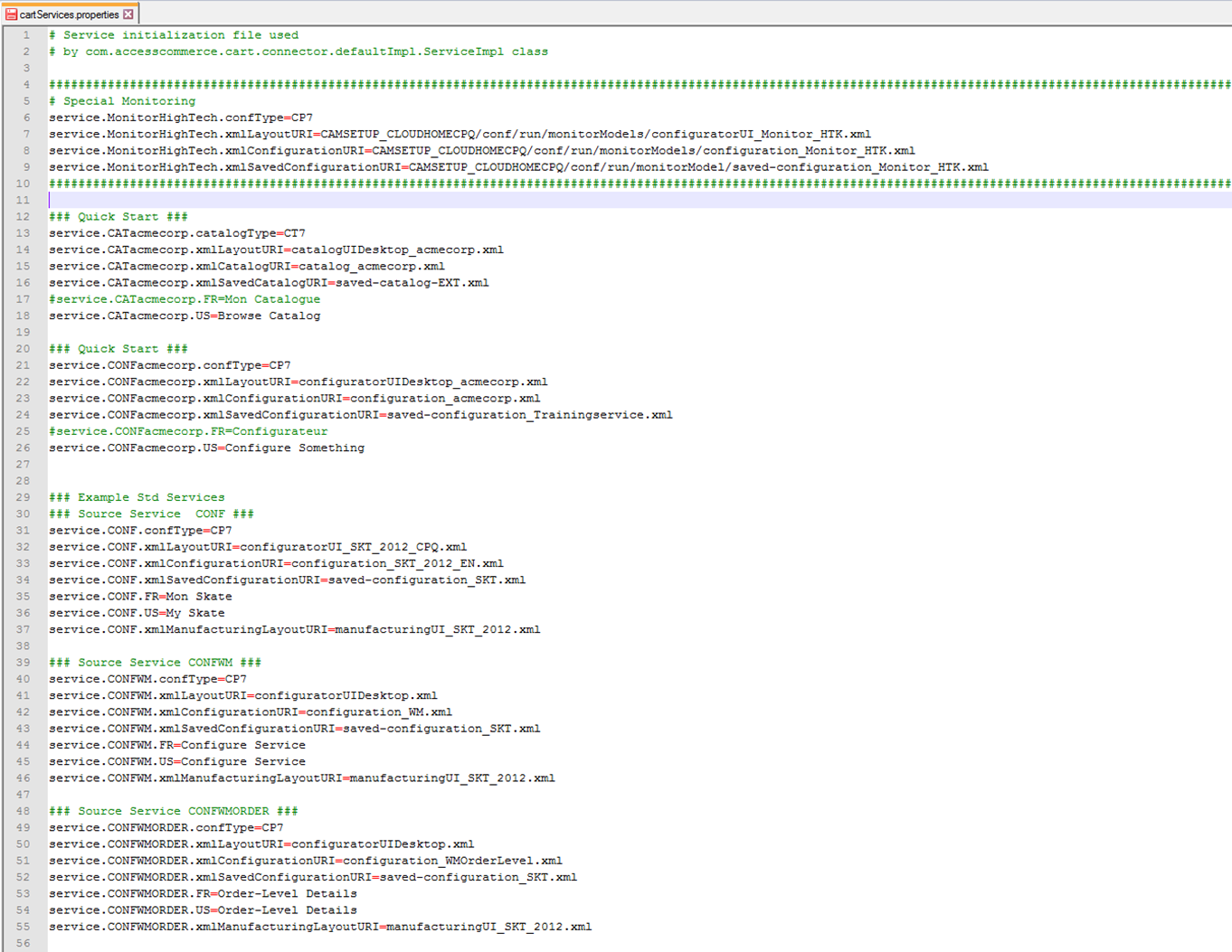
service.PROPOSAL.allowTableContentSelect=Yes

service.PROPOSAL.allowParagraphSelect=Yes

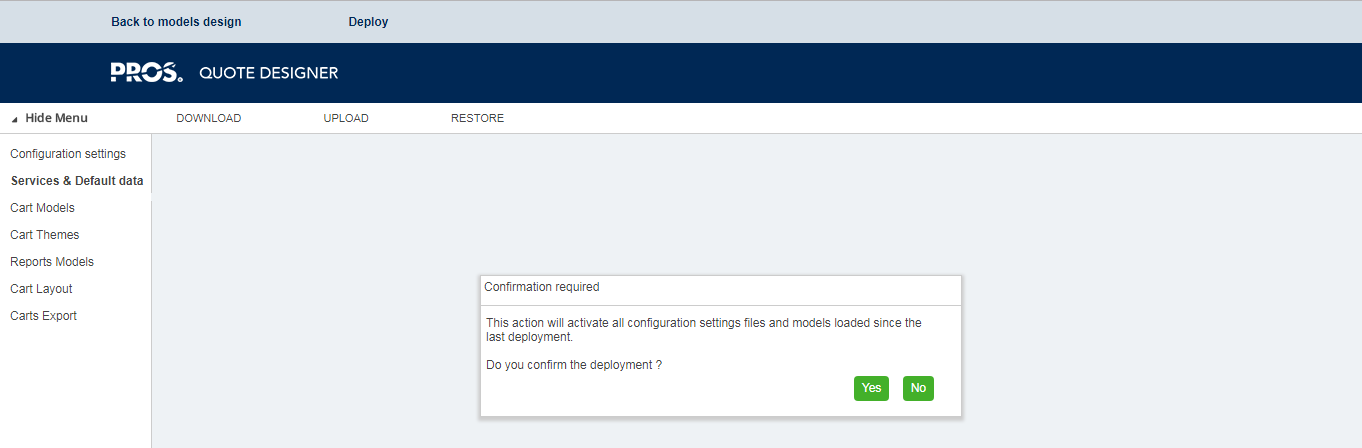
service.PROPOSAL.settingsAutoSave=Yes

service.PROPOSAL.keepLastDocumentOnly=Yes

**service.PROPOSAL.subDirectory=Proposal**



* Upload to **Services & Default Data**
* Click “Deploy” and “Yes”



## cartSettings.properties

Modify the cartSettings.properties file within the Quote Designer as follows:

* Download **cartSettings.properties** [within Quote Designer, under “Distribution / **Configuration Settings** / Download”]
* Update the userGroup settings below:

# User groups initialization file used

# by com.accesscommerce.cart.connector.defaultImpl.UserGroupManagerImpl class.

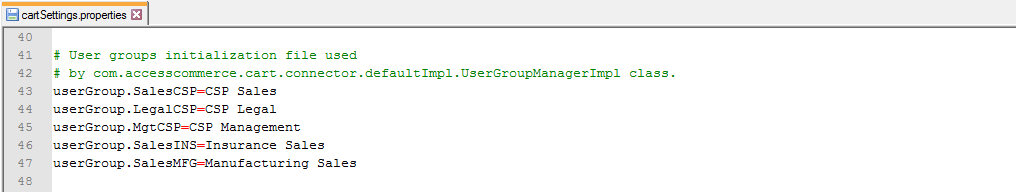
userGroup.SalesCSP=CSP Sales

userGroup.LegalCSP=CSP Legal

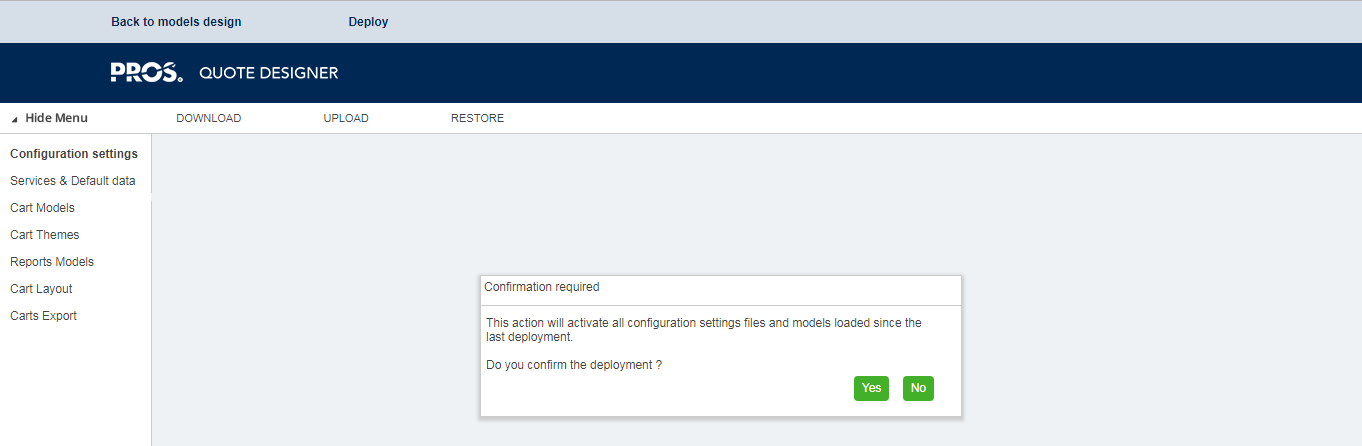
userGroup.MgtCSP=CSP Management

userGroup.SalesINS=Insurance Sales

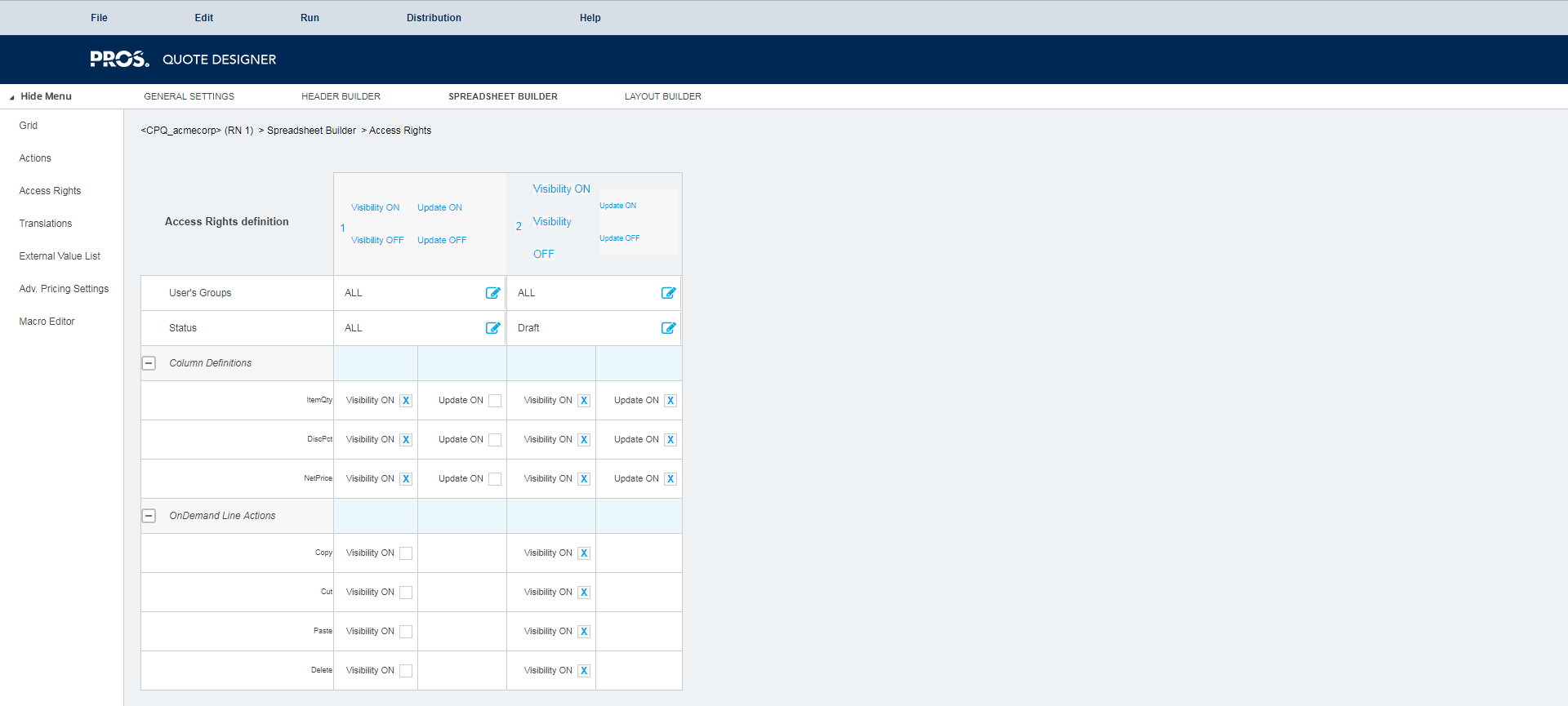
userGroup.SalesMFG=Manufacturing Sales



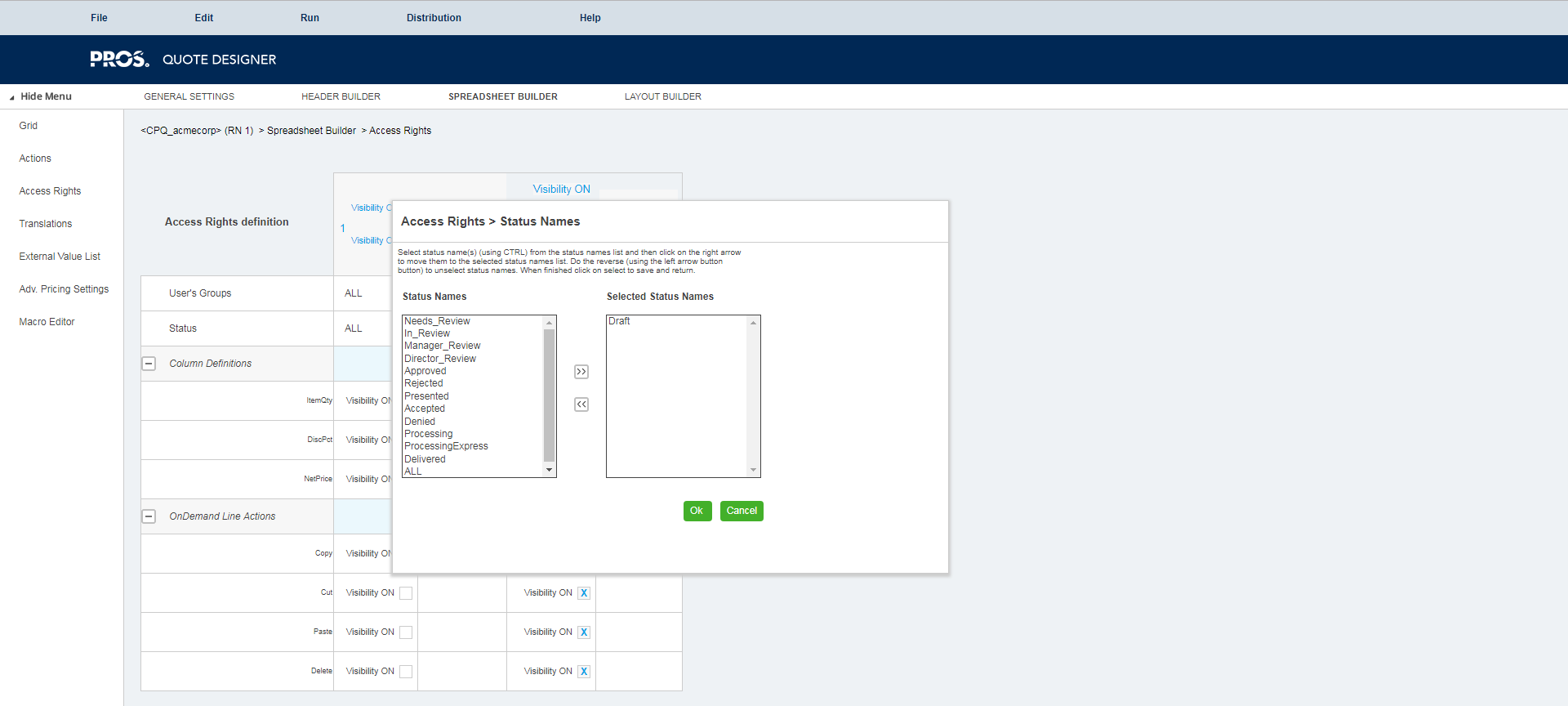
* Upload to **Configuration Settings**
* Click “Deploy” and “Yes”



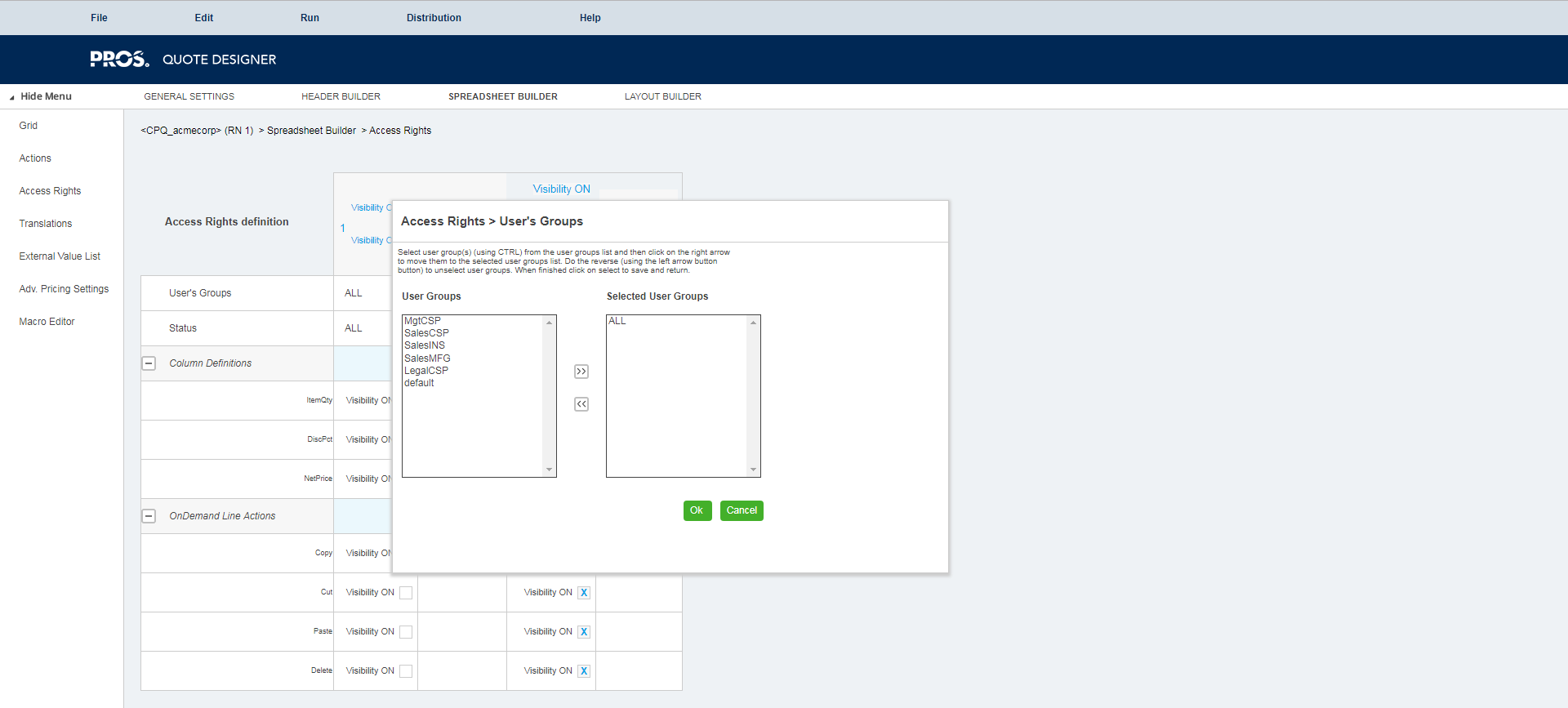
Note: The above user groups can be used for setting access rights with the cart model (note: the available column definitions and on-demand actions below are those that have been explicated set to ‘restricted’ [vs. ‘open’ or ‘hidden’] access):



Access rights in the current configuration are varied by status alone:



…But could also be varied by user group:



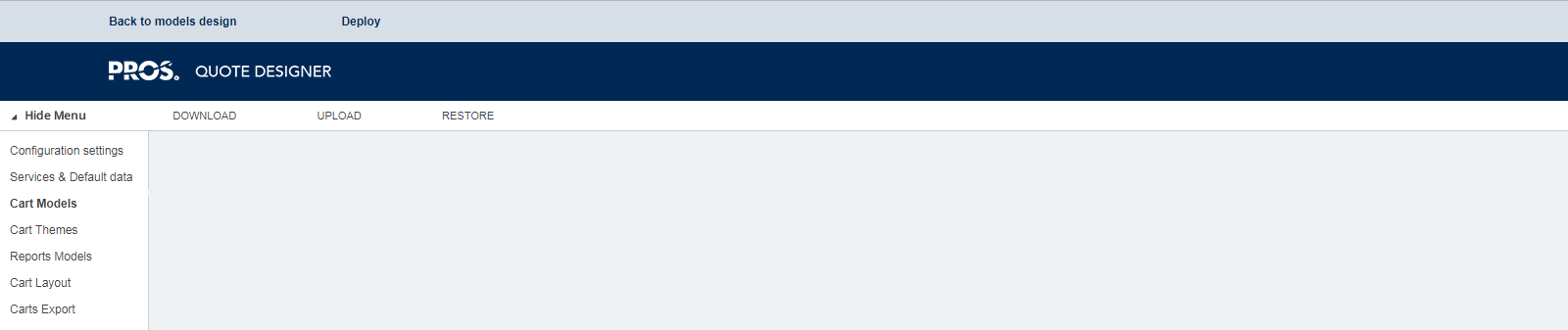
## cameleon.properties

Update the mail-related-data section of the **cameleon.properties** file on the server as follows:

# MAIL  
#   
# This configuration section contains MAIL-related data  
#======================================================================================================================================================================================================  
  
# Default address from mail  
mail.from=**CUSTOMER\_MAIL\_FROM**  
# Default address to mail  
#mail.to=**cameleon-ipad@cameleon-software.com**  
# Host server IP mail  
mail.server=**CUSTOMER\_MAIL\_SERVER**  
# Host server port mail, only used in case of mail server authenticated  
#mail.port=**587**  
# Protocol transport mail  
mail.protocol=**CUSTOMER\_MAIL\_PROTOCOL**  
# Username mail  
mail.user=**CUSTOMER\_MAIL\_USER**  
# Password mail  
mail.password=**CUSTOMER\_MAIL\_PWD**

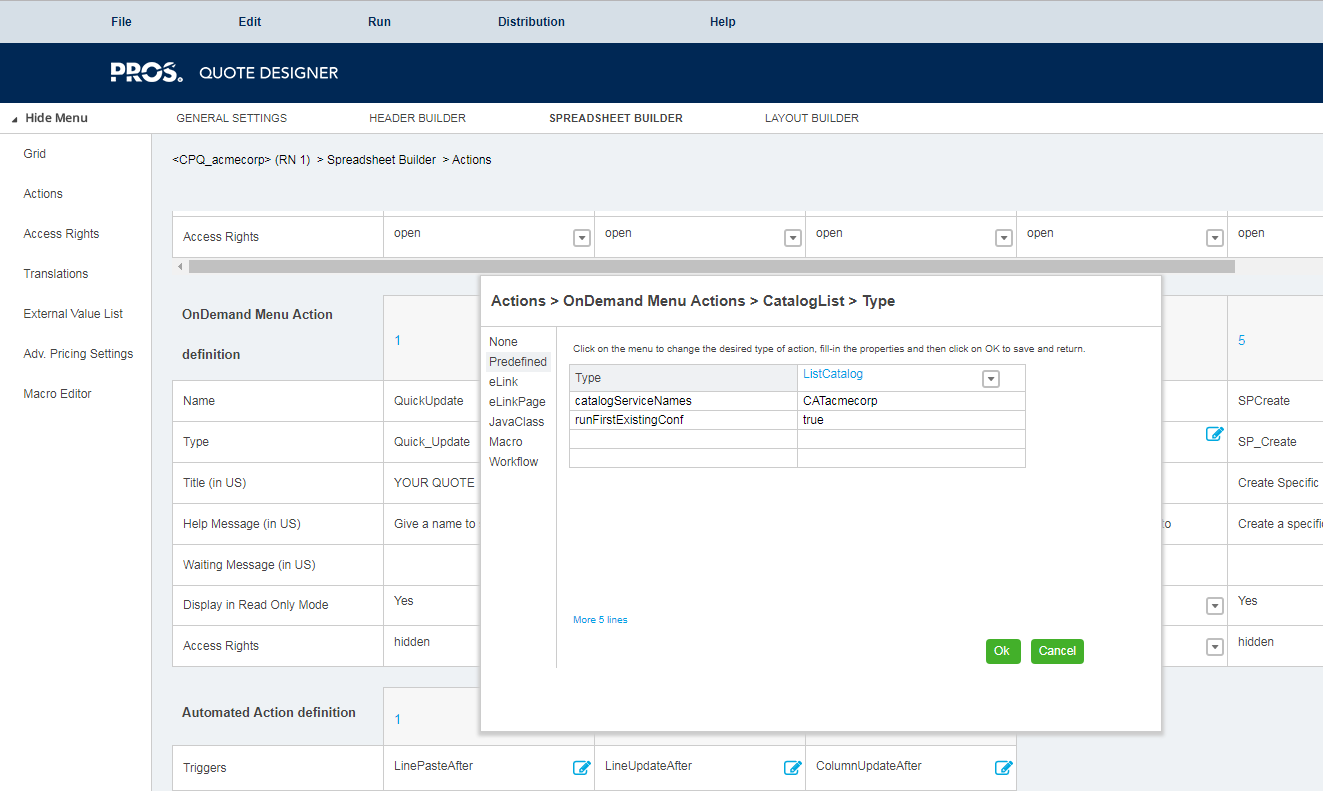
## Cart Model

**Upload** the included cart model ZIP into the Quote Designer via Distribution/Upload/Cart Models.

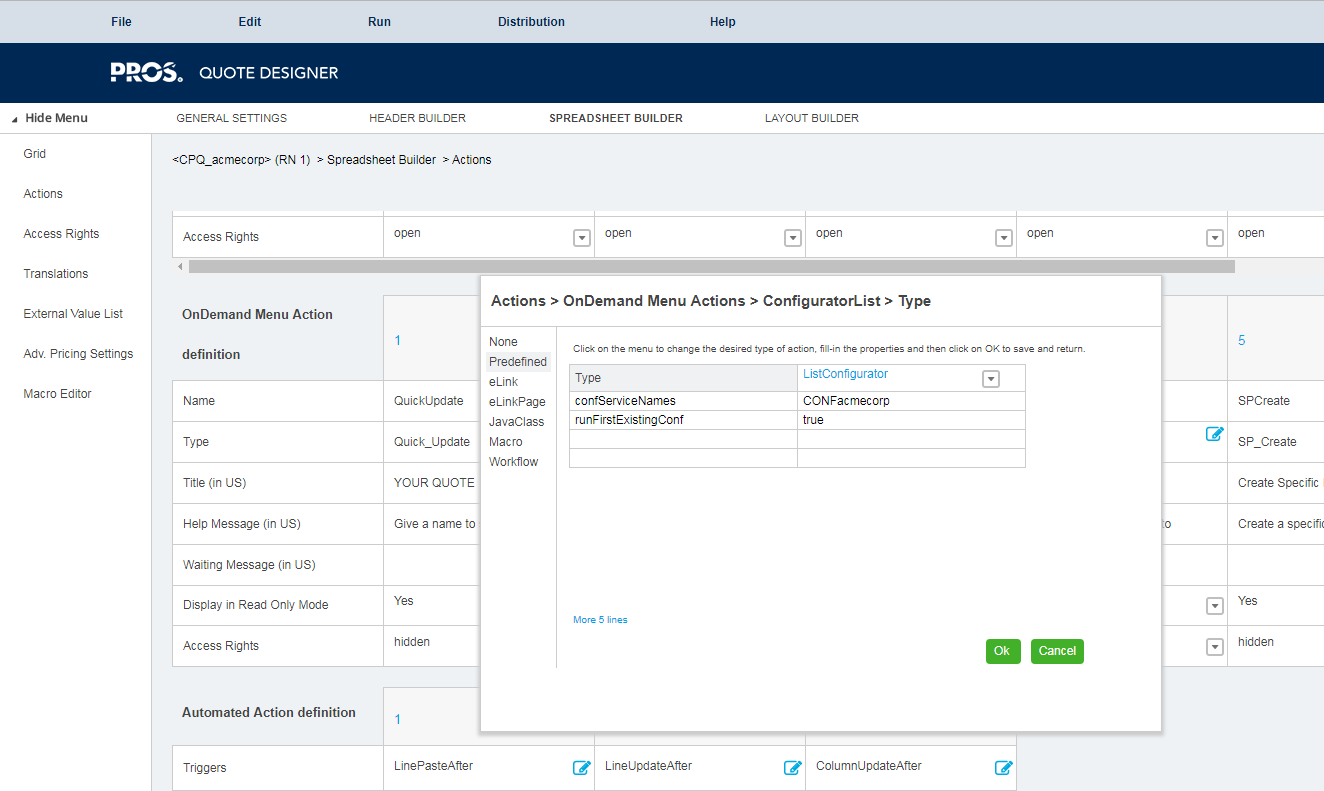


Then **modify** within the Designer as follows:

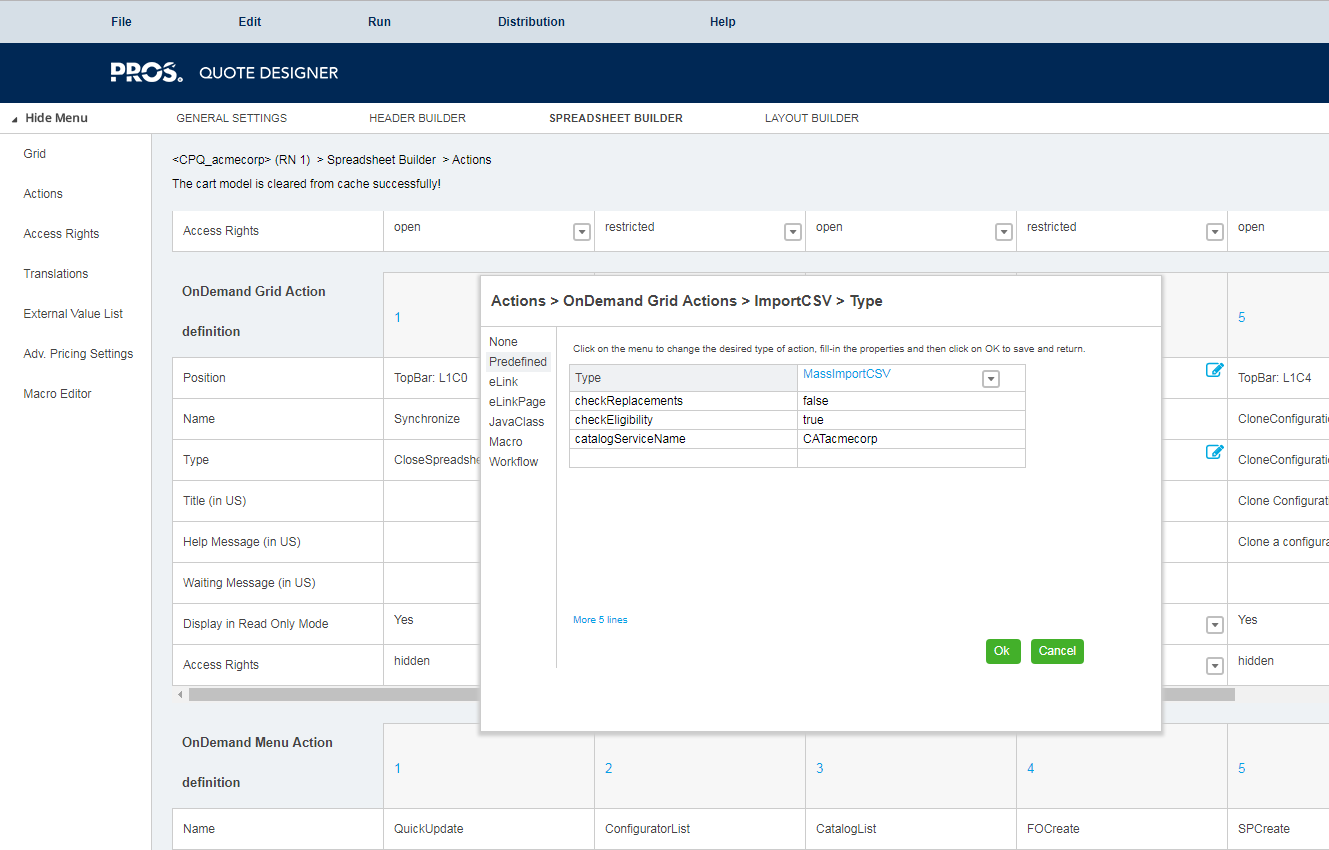
* **CPQ\_acmecorp.zip**
  + Open the Quote Designer
  + Open the “Default” cart model
  + Click “Distribution” and “Cart Models”
  + Upload CPQ\_acmecorp.zip
  + Save as ‘CPQ\_<Customer>’ (referenced in createcart\_<customer>.xml above)
  + Delete CPQ\_acmecorp cart model
  + On the **CatalogList** action, update catalogServiceNames to **‘CAT<customer>’** (defined in catServices.properties above)



* + On the **ConfiguratorList** action, update confServiceNames to **‘CONF<customer>’** (defined in catServices.properties above)



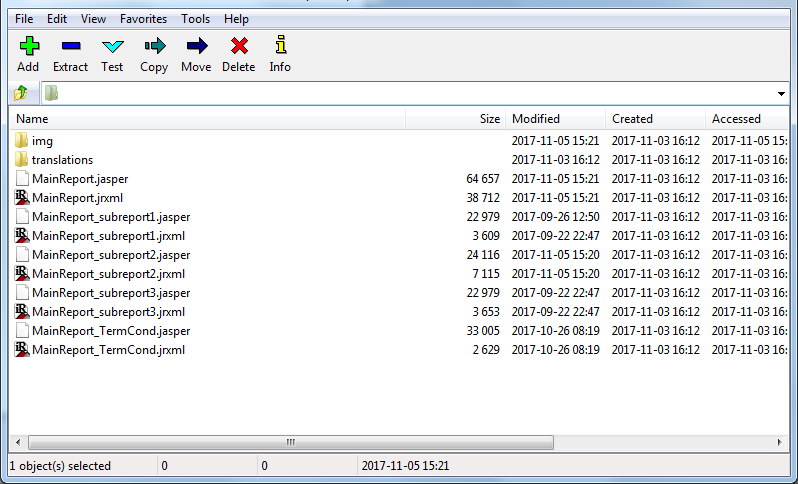
* + On the **ImportCSV** action, if not already done change the access from “hidden” to “open,” and change the type from “ImportCSV” to “**Mass ImportCSV**”
  + On the **ImportCSV** action, update catalogServiceName to **‘CAT<customer>’** (defined in catServices.properties above)
  + SAVE *and* DEPLOY



## Jasper Report

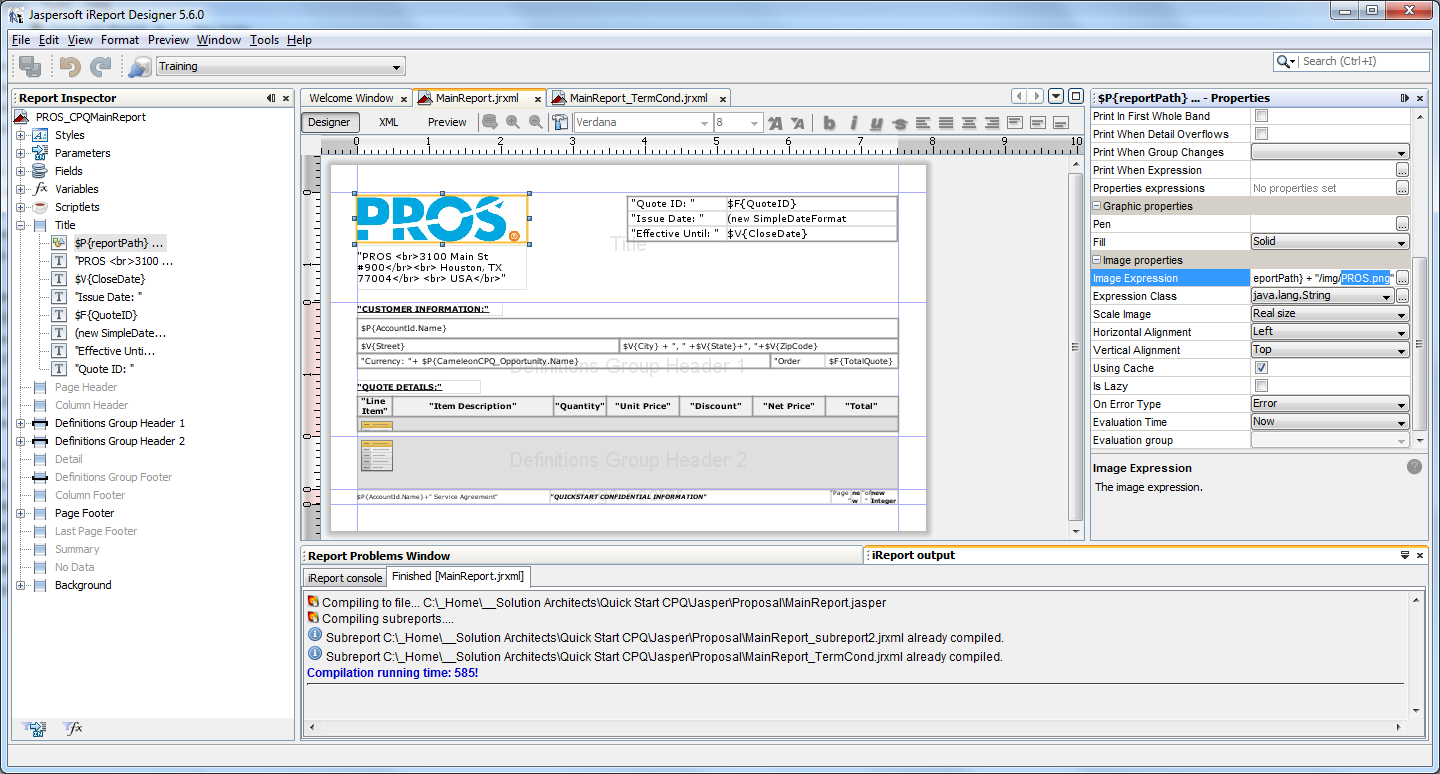
**Extract** the included Proposal.zip file into a file directory.

Replace PROS.png with **customer PNG or JPG logo** in the extracted \img folder of the Proposal.zip.

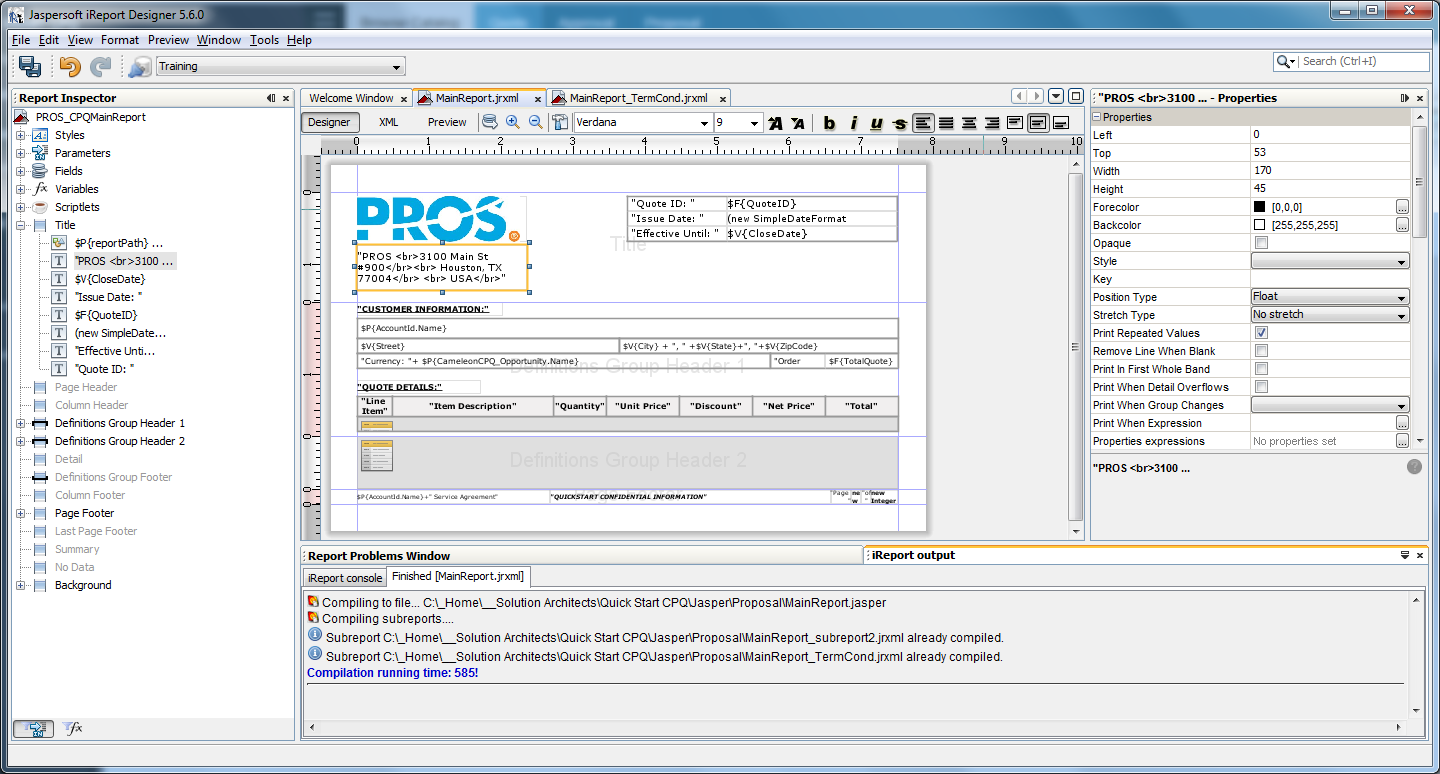


**Edit** the extracted **MainReport.jrxml** in Jaspersoft iReport Designer.

Change the name of the referenced logo file within the img directory of the zip file.

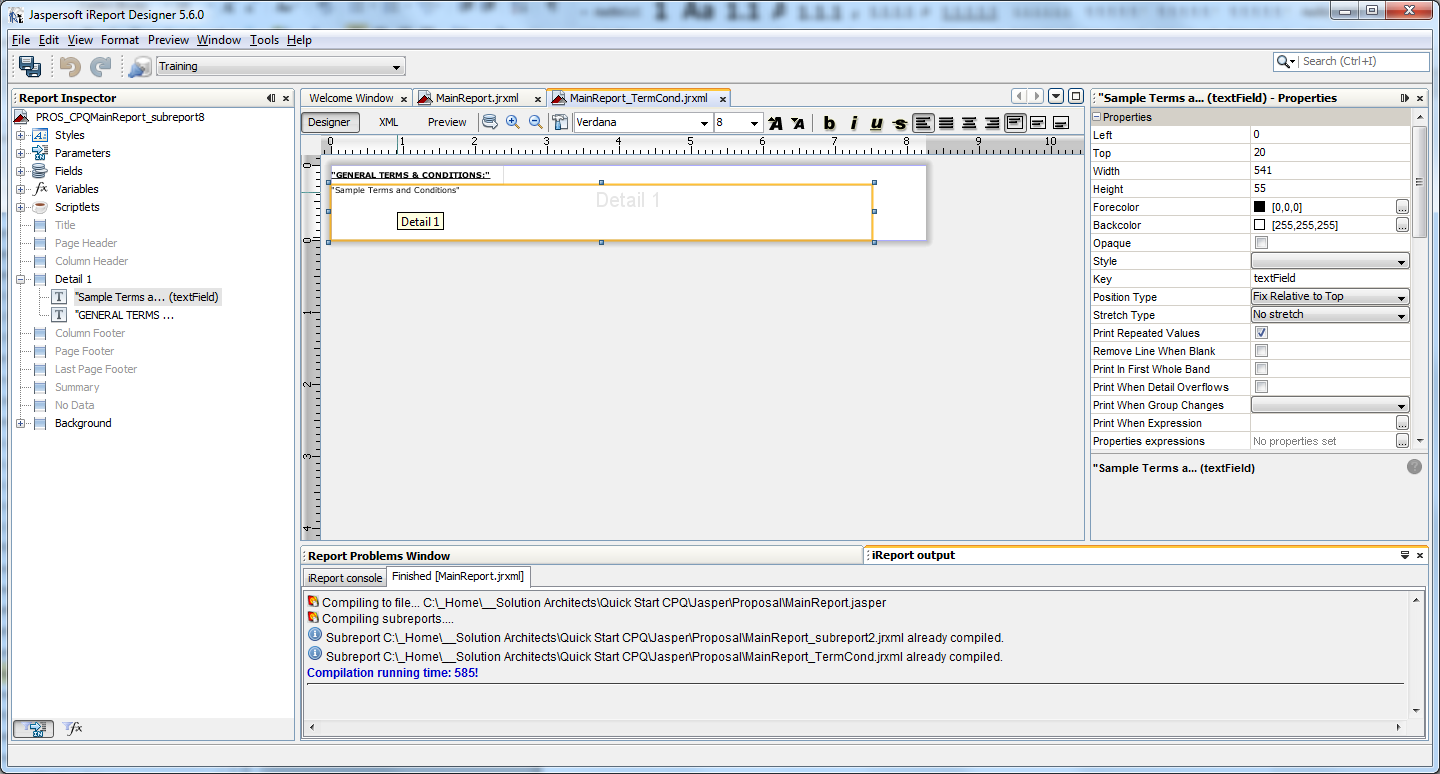


Update the hard-coded company address.



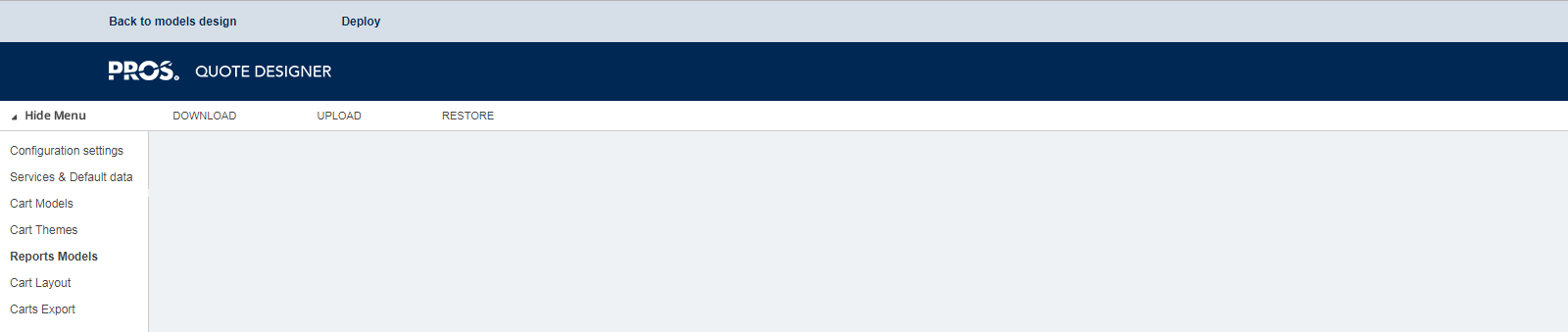
**Edit** the extracted **MainReport\_TermCond.jrxml** in JasperReports.

Update the hard-coded terms and conditions language.



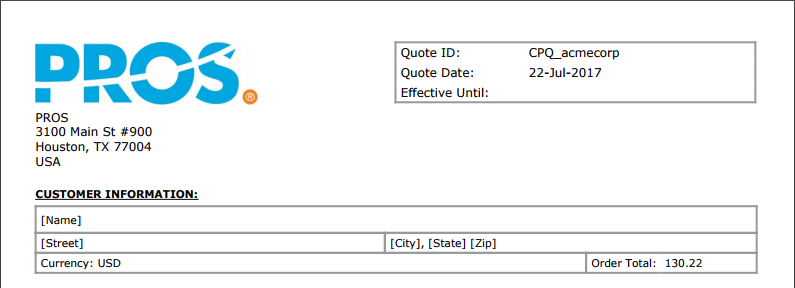
**Compile** the Jasper report and **create a new Proposal.zip**.

**Upload** the new Proposal.zip file into the Quote Designer via Distribution/Report Modules/Upload.



## Map-In Fields

**Validate** that the following are mapped in properly (and correct as necessary).

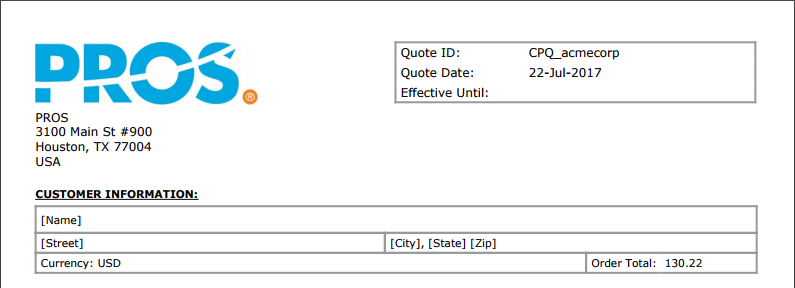


Within the Jasper report, the customer information fields reference the AccountId “implicit mapping object.” Within the “IN” mapping set in {Salesforce,Dynamics} the following fields must be explicitly selected:

* Customer Name: parameter AccountId.Name
* Customer Street: parameter AccountId.BillingStreet
* Customer City: parameter AccountId.BillingCity
* Customer State: parameter AccountId.BillingState
* Customer Zip: parameter AccountId.BillingPostalCode

## Header Fields

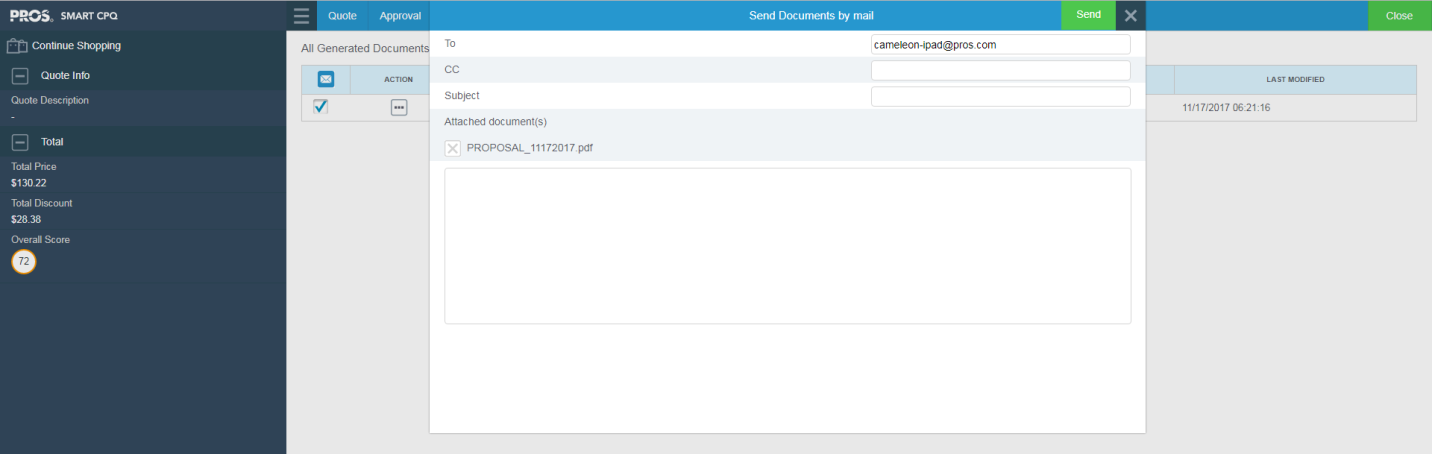
**Validate** that the following are properly configured in the cart model (and correct as necessary):



Within the Jasper report, the fields above reference the following Parameter, Change And Control, and Cart Info header fields in the cart model:

* Quote ID: /origin/form/cart/fields/ParameterTab.QuoteId
* Quote Date: /origin/form/cart/fields/ChangeAndControlTab.CreateDate
* Effective Until: /origin/form/cart/fields/CartInfoTab.CloseDate
* Currency: /origin/form/cart/fields/ParameterTab.Currency

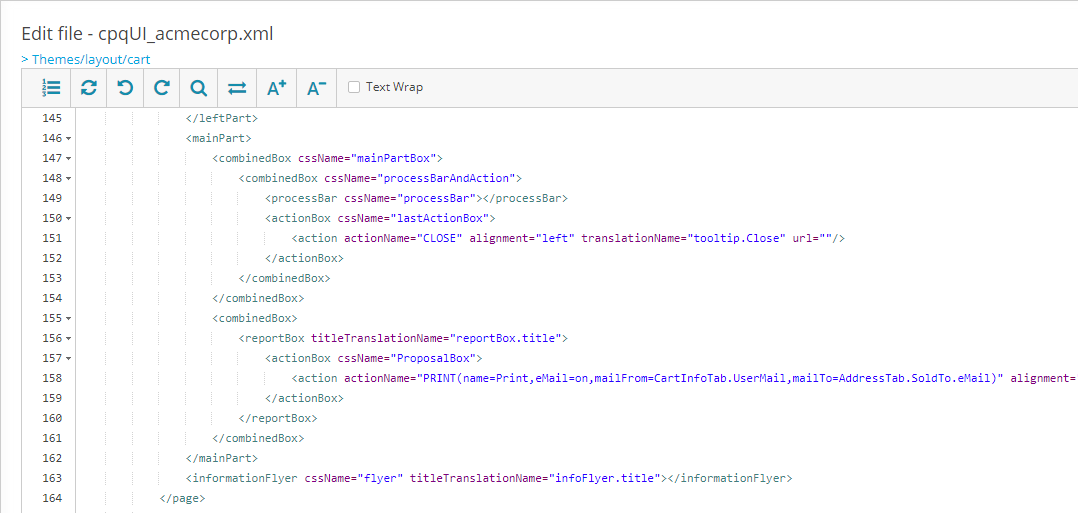
**Validate** that the proposal email from and to fields are mapped in properly (and correct as necessary):

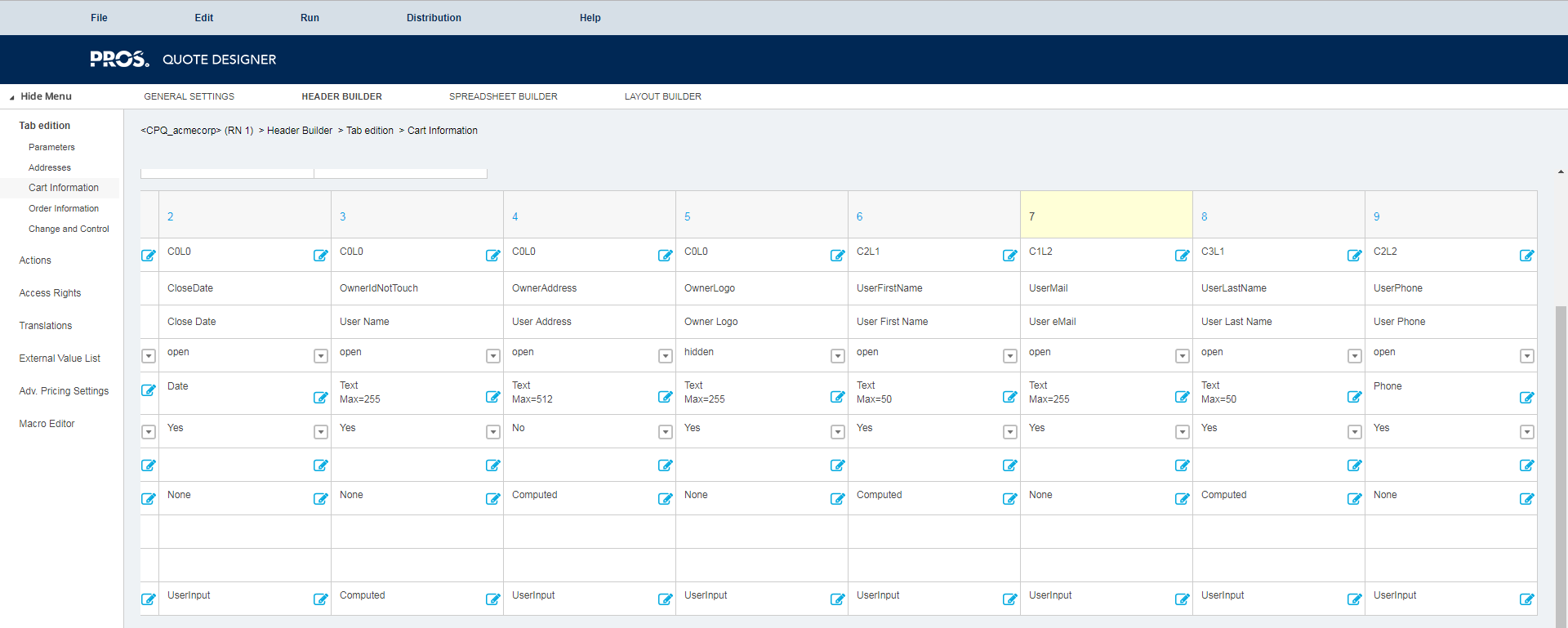


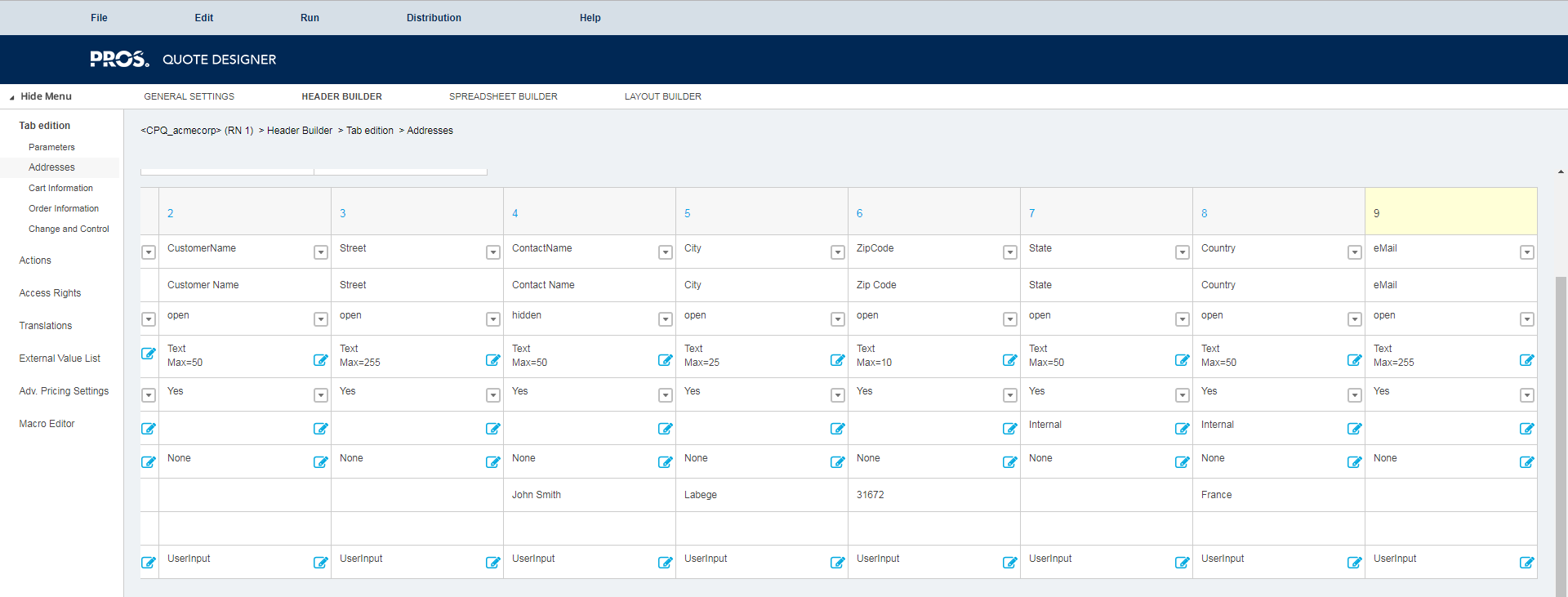
Within the cpqUI\_<customer>.xml theme file, the **email from** and **email to** fields reference the following Cart Info and Address header fields in the cart model:

* Email From: CartInfoTab.UserMail
* Email To: AddressTab.SoldTo.eMail

Note: Both the CartInfoTab.UserMail and AddressTab.SoldTo.eMail header fields are currently set to **user-input** and need instead to be **mapped in** from {Salesforce, Dynamics}.

****

****

****

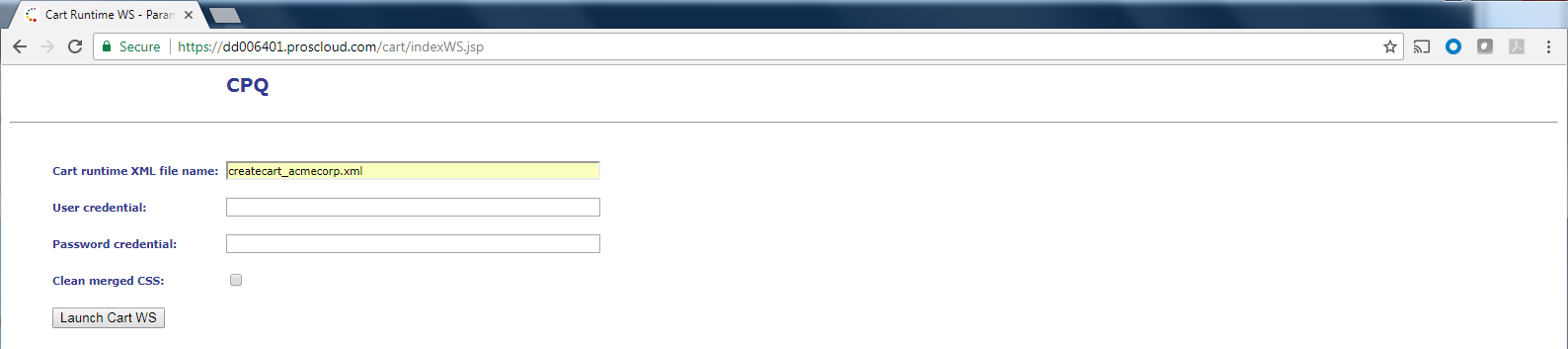
Default values for the from and to fields can also be configured in the **cameleon.properties** file:

# Default address from mail  
mail.from=**CUSTOMER\_MAIL\_FROM**  
# Default address to mail  
mail.to=**cameleon-ipad@cameleon-software.com**

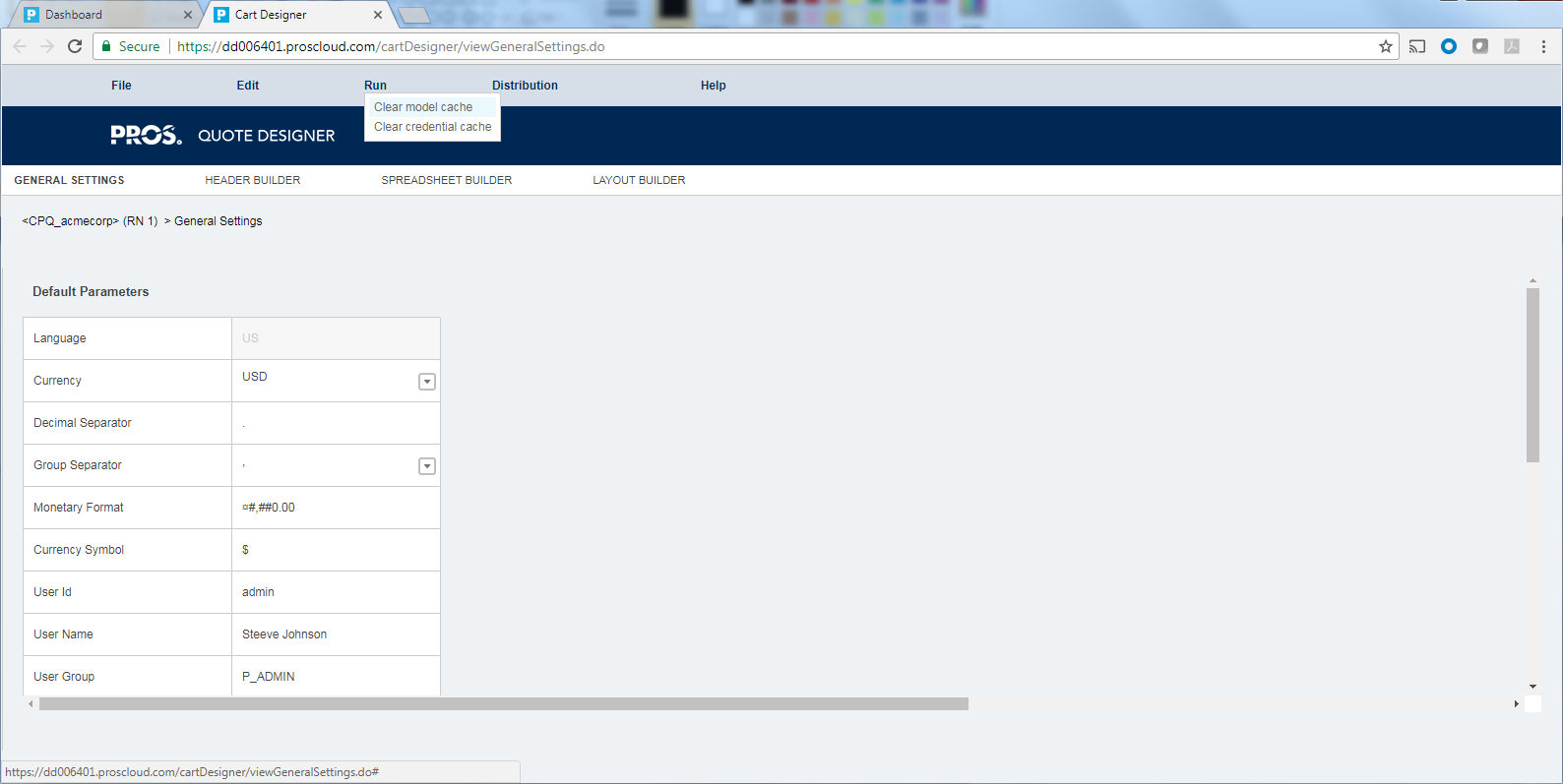
## Preview Cart

Preview the cart as follows:

* URL: [https://<???>.proscloud.com/cart/indexWS.jsp](https://dd006401.proscloud.com/cart/indexWS.jsp)
* File: createcart\_<customer>.xml (settings file above)
* Uid/pwd: \*\*\*\*\*\*\*\*/\*\*\*\*\*\*\*



If get error, clear the model cache:



# Appendix: Smart CPQ Catalog (CL) and Configuration Workflow (CP) Structure

## Catalogs

Root Catalog (CL) [note: a root catalog is itself a product collection]

└ Configuration Workflow (CP)

│ └:

└ Product Collection (CL)

│ └:

└Standard Item (SI)

└Sales Product (SP)

## Configuration Workflows

Root Configuration Workflow (CP)

└Configuration Workflow (CP)

│ └:

└Form (FO)

└Form Property (FP)

## Navigation

Root CP

└Cp A

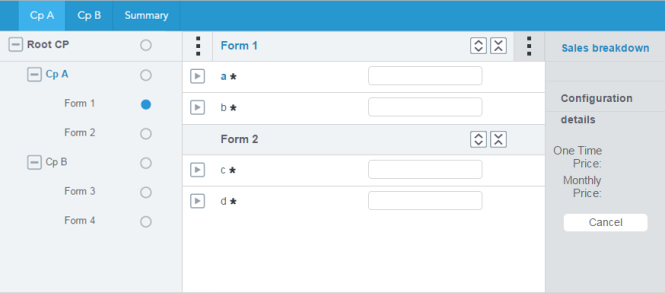
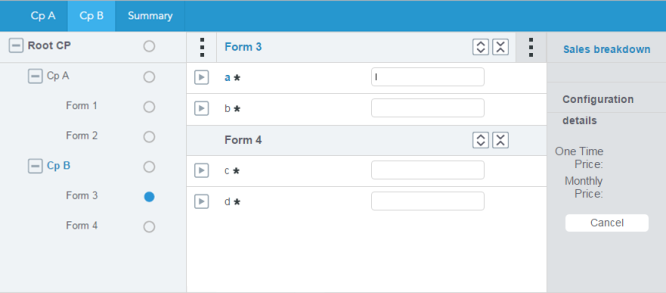
│ └Form 1

│ └Form 2

└Cp B

└Form 2

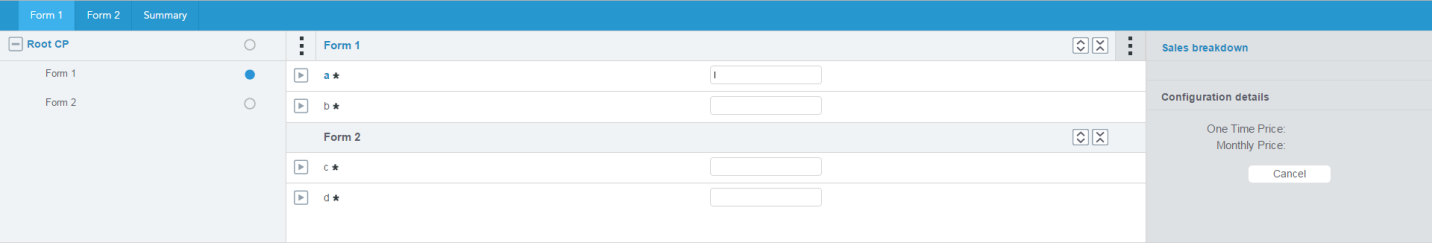
└Form 3

Root CP

└Form 1

└Form 2



## Form Properties

The input controls on each **form** within the **configuration process** [CP] are called **form properties**. Form properties can be dropdowns, radio buttons, check boxes, and other controls.

The values of each form property are stored in the **catalog** (as either “standard items” or “business values”) and are associated to the form properties via **BRCs** [Business Rule and Constraint objects].

Each form property has a **domain** (or “value selection”) BRC which specifies the set of values available for selection. A domain BRC may specify a subset of the catalog (a **product filter** BRC); a fixed set of values (a **matrix** BRC); or reference a **product link** (a product filter BRC with a product link). (A product link is used to dynamically set the domain of a form property based on the selected value of another form property.)

(BRCs can also be used to specify a default value, or to dynamically control when a form property exists and when it is required, visible, updatable, etc.)

# Version History

|  |  |  |  |
| --- | --- | --- | --- |
| **Change log** | | | |
| ***Version*** | ***Author*** | ***Date*** | ***Main changes & notes*** |
| 01 | Jason Day | 1/23/2019 | Initial document |
| 02 |  |  |  |
| 03 |  |  |  |
| 04 |  |  |  |