**eConcierge Canada Requirements**

**Revision History**

| Version No | Date | Prepared by / Modified by | Significant Changes |
| --- | --- | --- | --- |
| 0.1 | 15-May-19 | Oliver, Mark | * Initial draft |
| 0.2 | 17-May-19 | Oliver, Mark | * Several updates |
| 0.3 | 20-May-19 | Oliver, Mark | * Several updates |
| 0.4 | 21-May-19 | Oliver, Mark | * Initial version for review with HCL |
| 0.5 | 04-Jun-19 | Oliver, Mark | * Added some additional details about language localization approach for the Platform Portal and the Provider Portal |
| 0.6 | 11-Jun-19 | Oliver, Mark | * Added details on localization of serial number location to the Platform Portal |
| 0.7 | 12-Jun-19 | Oliver, Mark | * Closed some open questions in the Provider Portal section, Partner Portal section and the Customer Portal section |
| 0.8 | 25-Jun-19 | Oliver, Mark | • Added some additional details on shopping cart pages to the Partner Portal section and Customer Portal section |
| 0.9 | 01-Aug-19 | Oliver, Mark | * Added Bambora certified interface details to the Platform Portal section |
| 1.0 | 05-Aug-19 | Oliver, Mark | * Added new Partner ordering validation checks to the Partner Portal section |
| 1.1 | 06-Aug-19 | Oliver, Mark | * Added details on configuring Bambora credentials to the Partner Portal section |
| 1.2 | 10-Aug-19 | Oliver, Mark | * Added details on Bambora integration to the Customer Portal section and the Payment section |
| 1.3 | 13-Aug-19 | Oliver, Mark | * Updated details and closed open questions in the Distributor section |
| 1.4 | 14-Aug-19 | Oliver, Mark | * Added removal of sitemap to the Partner Portal section * Added additional error messaging details to the Payment section |
| 1.5 | 15-Aug-19 | Oliver, Mark | * Added additional details related to answers received from Bambora to the Payment section |
| 1.6 | 30-Aug-19 | Oliver, Mark | * Removed entry of email address in the Customer Portal section * Changed source of email address in the Payment section * Updated logic for decline versus error response handling in the Payment section * Added details on change to Order Placed notification to the Notifications section |
| 1.7 | 12-Sep-19 | Oliver, Mark | * Added details for creating the payment record for pre-auth completion decline scenarios in the Payment section |

**Table of Contents**

[High Level Requirement 3](#_Toc9262447)

[Assumptions 3](#_Toc9262448)

[UI Requirements 3](#_Toc9262449)

[Platform Portal 3](#_Toc9262450)

[Provider Portal 6](#_Toc9262451)

[Partner Portal 10](#_Toc9262452)

[Customer Portal 16](#_Toc9262453)

[Supplies Assistant 21](#_Toc9262454)

[Other Requirements 22](#_Toc9262455)

[Notifications 22](#_Toc9262456)

[Reports 23](#_Toc9262457)

[Integration 23](#_Toc9262458)

[Distributor 23](#_Toc9262459)

[Tax 24](#_Toc9262460)

[Service 24](#_Toc9262461)

[Payment 25](#_Toc9262462)

[Functional Spec 38](#_Toc9262463)

[Wireframes 38](#_Toc9262464)

[Approach Document 39](#_Toc9262465)

[Configuration 39](#_Toc9262466)

[Open Questions 39](#_Toc9262467)

## High Level Requirement

Expand the existing eConcierge offering to Canada, with initial focus on the VAR program but also ensuring support for eventual expansion to the NP program as well

## Assumptions

1. A new Canada Provider and Program(s) will be defined rather than trying to re-use those that were defined as part of the original expansion to Canada
2. All Partner and Customer facing content will need to be localized
3. All Partner and Customer initiated financial transactions and display of financial amounts will be in Canadian dollars, even if not explicitly shown with a currency description of Canadian Dollar or a symbol of CAD
4. All existing Partner and Customer features enabled for the US program will potentially also be enabled for the Canada program
5. A goal of the implementation should be to re-use existing functionality, UI, database structure, etc. where feasible, in order to reduce the overall development and testing effort

## UI Requirements

### Platform Portal

1. Language Localization
   1. Platform facing content does not need to be localized for French, but some localization related changes will be needed to support language localization for lower level Portals (see details under UI Changes)
2. Currency Localization
   1. No changes should be required as there are no financial details shown in the Platform Portal
3. Address Localization
   1. All address details should be shown in a Canadian address format if they are associated to a country of “Canada”, including
      1. Use of “Postal Code” instead of “Zip Code”
      2. Use of “Province” instead of “State”
   2. Note: Above should happen by default as the Platform Portal already supports entering and displaying Canadian addresses
4. UI Changes
   1. Login Page
      1. Add the ability for user to select a language
         * English
         * French (Français)
      2. Language choice should be “sticky” across sessions
      3. Selection of a language for a Platform Portal user would determine the content shown on the login page, but have no impact after login since the Platform Portal is not being localized for language
   2. Catalog -> Models
      1. Add the ability to enter and view values in additional languages for the following Model related configuration fields
         * Description
         * Additional Information
         * Serial Number Location
         * Note: Approach should follow the same approach as used for configuring in multiple languages during the previous Canada implementation
      2. Schema Impact
         * X\_DEVICE\_MODEL
           + Description (move field and existing data to corresponding locale table)
           + Additional Information (move field and existing data to corresponding locale table)
           + Serial Number Location (move field and existing data to corresponding locale table)
         * X\_DEVICE\_MODEL\_LOCALE (New table)
           + ID (unique)
           + Device Model ID (foreign key)
           + Description
           + Additional Information
           + Serial Number Location
           + Locale ID (foreign key)
           + Standard who/when fields
   3. Settings -> Configuration Settings
      1. Add new section to configure Vertex credentials for Canada
         * Section Title = “eConcierge Canada - Tax System Credentials”
         * Field to configure User ID
           + Field label = “User ID”
           + Free form text
           + Required

User should be shown a validation error message if trying to save settings without entering a value

Message = “Please provide data in Canada Tax System User ID.”

* + - * + Visible on view/edit pages
      * Field to configure Password
        + Field label = “Password”
        + Free form text
        + Required

User should be shown a validation error message if trying to save settings without entering a value

Message = “Please provide data in Canada Tax System Password.”

* + - * + Masked for input and viewing
        + Visible on view/edit pages
      * “Test” button is required
      * Positioned below existing “eConcierge United States - Tax System Credentials” section
        + Note: If it would reduce development effort, the existing “FCP United States - Tax System Credentials” section could be re-purposed for this rather than adding a new section

1. UI Configuration
   1. Providers -> Providers
      1. Define a new Provider
         * Definition details
           + Provider Name = “Canada eConcierge”
           + Provider Type = “Canada eConcierge”
           + Description = “Canada eConcierge”
           + Country = “Canada”
           + Certified Interfaces

Payment Gateway = “Bambora”

Tax Engine = “Vertex Tax Engine - Canada”

* + - * + User ID = “can\_econ\_admin”
  1. Providers -> Provider Reports
     1. Add report(s) to new Provider as needed
        + Same reports as used for the existing NARS provider should be configured for the Canada Provider
  2. Catalog -> Models
     1. Add additional Model details in French as needed
  3. Catalog -> Consumables -> Categories
     1. Add additional Consumable Category details in French as needed
  4. Catalog -> Consumables -> Thresholds
     1. No configuration changes should be required
  5. Catalog -> Consumable Part #
     1. Add additional Consumable Part details in French as needed
  6. Catalog -> SKU Mapping
     1. Update Disti SKUs for “SYNNEX Canada” sourcing location as needed
  7. Setup -> Base Data
     1. No configuration changes should be required
        + Note: Previously configured country, provinces, and language for Canada will be used
  8. Setup -> Certified Interfaces
     1. Define a new Certified Interface
        + Name = “Bambora”
        + Key = “BAMBORA\_PAYMENT\_GATEWAY”
        + Type = “Payment Gateway”
        + Country = “CA”
     2. Previously configured Distributor System and Tax Engine interfaces will be used as follows
        + Synnex Distributor Interface - Canada
        + Vertex Tax Engine – Canada
     3. Schema Impact
        + X\_CERTIFIED\_INTERFACE
          - Seed new value for Bambora
  9. Setup -> Sourcing Locations
     1. Configure Promotions, SME, and Device Purchasing enablement values for the existing “SYNNEX Canada” sourcing location
  10. Settings -> Configuration Settings
      1. Configure Vertex credentials for Canada
         + User = hclusers\_prod
         + Password = *same password as US*

1. Questions
   1. Open
      1. What should the settings for Promotions, SME, and Device Purchasing enablement on the SYNNEX Canada sourcing location be?

**Answer:** TBD. Note: Answer will be based on what features the Canada business chooses to enable or not enable.

* 1. Closed
     1. What Provider reports will be used for the new Canada Provider?

**Answer:** As a start, will just use the same reports as used for the existing US (NARS) Provider.

* + 1. Will a new Payment Gateway Certified Interface need to be defined?

**Answer:** Yes. A new certified interface for Bambora will need to be defined.

* + 1. What will the Vertex credentials for Canada be?

**Answer:** Same credentials as used for the US will be used for Canada.

* + 1. What approach should be used for localizing Partner and/or Customer facing content that is currently stored in the Lookup (X\_LOOKUP) table?

**Answer:** A new table (X\_LOOKUP\_LOCALE) will be created to store Lookup data that needs to be localized. Examples of this data from the Platform Portal are the “Color Capability” and “Function” values configured for a Device Model that are shown in the Customer Portal.

### Provider Portal

1. Language Localization
   1. Provider facing content does not need to be localized for French, but some localization related changes will be needed to support language localization for lower level Portals (see details below under affected pages)
2. Currency Localization
   1. All financial details shown should be in Canadian dollars, using the same currency symbol and display format as currently used for US dollars
   2. Note: Above should happen by default due to the Provider being associated to Canada in the Platform Portal, so should not require any development work
3. Address Localization
   1. All address details should be shown in a Canadian address format, including
      1. Use of “Postal Code” instead of “Zip Code”
      2. Use of “Province” instead of “State”
      3. Country shown as “Canada” (where applicable)
   2. Note: Above should happen by default due to the Provider being associated to Canada in the Platform Portal, so should not require any development work
4. UI Changes
   1. Login Page
      1. Add the ability for user to select a language
         * English
         * French (Français)
      2. Language choice should be “sticky” across sessions
      3. Selection of a language for a Provider Portal user would determine the content shown on the login page, but have no impact after login since the Provider Portal is not being localized for language
   2. Programs -> Program Definition -> Banners
      1. Add the ability to enter values or upload files in additional languages for the following banner configuration fields
         * Banner Image
         * Alt Text
         * Modal Title Text
         * Modal Header Text
         * Modal Body Details
         * Note: Approach should follow the same approach as used for configuring in multiple languages during the previous Canada implementation
      2. Schema Impact
         * X\_BANNER\_ELEMENTS
           + Image File Name (move field and existing data to corresponding locale table)
           + Image Content (move field and existing data to corresponding locale table)
           + Alt Text (move field and existing data to corresponding locale table)
         * X\_BANNER\_MODALS
           + Title (move field and existing data to corresponding locale table)
           + Heading (move field and existing data to corresponding locale table)
           + Body (move field and existing data to corresponding locale table)
         * X\_BANNER\_ELEMENTS\_LOCALE (New table)
           + ID (unique)
           + Banner Elements ID (foreign key)
           + Image File Name
           + Image Content
           + Alt Text
           + Locale ID (foreign key)
           + Standard who/when fields
         * X\_BANNER\_MODALS\_LOCALE (New table)
           + ID (unique)
           + Banner Modals ID (foreign key)
           + Title
           + Heading
           + Body
           + Locale ID (foreign key)
           + Standard who/when fields
   3. Partners -> Add Partner
      1. Add the ability to select Payment Methods
         * Open Account
         * Credit Card
         * Note: Implementation of this should be the same as was previously done for US VAR Partners
   4. Promotions -> List of Promo Coupons
      1. Add the ability to enter and view value in additional languages for the following promotion configuration field
         * Additional Details
         * Note: Approach should follow the same approach as used for configuring in multiple languages during the previous Canada implementation
      2. Schema Impact
         * X\_PROVIDER\_PROMO\_COUPONS
           + Additional Details (move field and existing data to corresponding locale table)
         * X\_PROVIDER\_PROMO\_COUPONS\_LOCALE (New table)
           + ID (unique)
           + Provider Promo Coupons ID (foreign key)
           + Additional Details
           + Locale ID (foreign key)
           + Standard who/when fields
   5. Tools
      1. Update “Resubmit Order” tool as needed to handle Canada orders
5. UI Configuration
   1. Programs -> Program Definition -> Basic Details
      1. Define a new Program
         * Name = “Canada - Xerox eConcierge”
         * Description = “Xerox eConcierge for Canada”
         * Program Type = “VAR”
         * Program Category = “Supplies”
         * Countries = “Canada”
         * Sourcing Locations = “SYNNEX Canada”
         * Offer Replacement Parts = No
         * Order by Model = Yes
         * Allow Partner to Place Orders = Yes
         * Allow Promotions = TBD
           + Enable New Customer Promotion = TBD
           + Customer Promo Code = TBD
         * Check for Duplicate Orders = Yes
           + Check for Same Device = Yes
           + Order Time Difference = 600
         * Allow to add Meter Toner Enabled devices = TBD
         * Enable Backorder = Yes
         * Order Bean Life Period = 15
         * Device not reporting threshold = 7
         * Device data update waiting time = 2
         * Non-Monitored Devices Enabled = TBD
         * SA Warning Message at Time of Login Enabled = Yes
           + Prevent Login if SA Not Latest Version = No
         * Send device not reporting email alert = Yes
           + Device not reporting for = 17
           + Repeat device-not-reporting email = 3
           + Repeat device-not-reporting email after = 14
         * Send Agent installation reminder email = Yes
           + Agent not installed after customer registration for = 7
           + Repeat Agent-installation-reminder email = 3
           + Repeat Agent-installation-reminder email after = 7
         * Send device not added reminder email = Yes
           + Device not added after Agent installation for = 7
           + Repeat device-not-added reminder email = 3
           + Repeat device-not-added reminder email after = 7
         * Stock Check Retry Configuration = Yes
           + Stock Check Retry = 1
         * Eligible Payment Options = “Buy Now”
         * Support Email = TBD
         * Support Phone Number = TBD
         * Remove / Re-add Grace Period = 14
         * Turn On Survey = No
           + Survey URL = *blank*
         * Device Purchasing Enabled = TBD
           + Send Device Not Connected Email = TBD
           + Threshold Days = TBD
           + Number of Repeats = TBD
           + Repeat After Days = TBD
         * All other configuration can use default values
   2. Programs -> Program Definition -> T & C
      1. Upload Partner and End User T & C for new program in English and French
   3. Programs -> Program Definition -> Notifications
      1. Configure System, Partner, and Customer notifications for new program in English and French
   4. Programs -> Program Definition -> Help Docs
      1. Upload Partner, Customer, and Supplies Assistant Help Docs for new program in English and French
   5. Programs -> Program Definition -> Identify Partner Reports
      1. Select Partner reports for new program
   6. Programs -> Shipping & Handling -> Shipping Methods
      1. Configure shipping methods for new program in English and French
   7. Programs -> Models and Parts -> Eligible Models
      1. Add applicable models and default part overrides for new program
      2. Note: Only Xerox models will be added to the program
   8. Programs -> Models and Parts -> Consumable Part #
      1. Add applicable consumable parts for new program
   9. Programs -> Models and Parts -> Consumables and Thresholds
      1. Configure consumable and threshold levels for new program
   10. Programs -> Loyalty Benefit Configuration -> Loyalty Message
       1. Configure service loyalty message for new program in English and French
   11. Programs -> Loyalty Benefit Configuration -> Loyalty Thresholds
       1. Configure service loyalty thresholds for new program
   12. Base Data -> Base Prices -> Consumable Parts
       1. Upload price list with Canadian specific Disti Costs and currency
   13. Base Data -> Credit Limit
       1. Configure credit limit in Canadian currency as needed
   14. Accounts -> Users
       1. Create additional users as needed
6. Questions
   1. Open
      1. What should be used for the TBD values when creating the Program?

**Answer:** TBD. Note: Most of the TBD items will be based on what features the Canada business chooses to enable or not enable.

* 1. Closed
     1. Will localization be required for the customer survey URL that is defined at the Program level?

**Answer:** No. This feature will not be used for the initial implementation. This can be re-visited if it is decided to use it at some point in the future.

### Partner Portal

1. Language Localization
   1. All Partner facing content needs to be localized for French, including
      1. Tab titles
      2. Menu titles
      3. Column titles
      4. Field labels
      5. Button labels
      6. Link labels
      7. Selection choices in dropdown boxes
      8. Other static content, such as overview/explanatory text and error/warning messages
      9. Variable content that is localized via configuration in the Platform and Provider Portals
      10. Marketing collateral links shown in the setup wizard and content that is opened via the links
      11. Customer links shown in the setup wizard and on the Branding page, including the Customer Registration page link, the Supplies Assistant download link, and the Customer Portal link
      12. Header and Footer links and content that is opened via the links
      13. Downloaded files and content
      14. Report names and content
      15. All other Partner facing content not specifically mentioned above, with the exception of data entered directly by the Partner in the Partner Portal via input fields
   2. Language localization should include all eConcierge versions of the Partner Portal, including
      1. VAR
      2. NP
      3. NP+
   3. No localization of logos/branding is required
   4. Language localization approach should follow the same approach as used for the Partner Portal localization during the previous Canada implementation
      1. Note: Any new Partner facing feature content that was not in place during the previous localization effort also needs to be included in the current effort
2. Currency Localization
   1. All financial details shown should be in Canadian dollars, using the same currency symbol and display format as currently used for US dollars
   2. Note: Above should happen by default due to the related Provider being associated to Canada in the Platform Portal, so should not require any development work
3. Address Localization
   1. All address details should be shown in a Canadian address format, including
      1. Use of “Postal Code” instead of “Zip Code”
      2. Use of “Province” instead of “State”
      3. Country shown as “Canada” (where applicable)
   2. Note: Above should happen by default due to the Provider being associated to Canada in the Platform Portal, so should not require any development work
4. UI Changes
   1. Login Page
      1. Add the ability for user to select a language
         * English
         * French (Français)
      2. Language choice should be “sticky” across sessions
      3. Language choice should update the user’s language preference in the DB after log in (if different than the current preference)
         * Note: Language preference for a US Partner user should not be updated in the DB after login, even if it is different than their current preference. In other words, the language choice for a US Partner user as stored in the DB should only ever be English.
      4. Language choice should determine content shown to Partner on the login page and after log in (as listed above under Language Localization)
         * Note: Language choice should have no impact for a US Partner user after login. Their experience should be exactly the same as it currently is (everything shown in English), regardless of what language choice they may have made on the Login page.
   2. Setup Wizard
      1. Account
         * Add Tax Registration fields for Canada under existing “Tax Setup” section
           + GST/HST Registration ID
           + QST Registration ID
           + Note: Implementation of these fields should be the same as was previously done on the Payment step, including only requiring entry of “QST Registration ID” for Partners with a Quebec address
      2. Pricing
         * Currency should default to Canadian Dollar and not be updateable
           + Note: This should happen by default based on configuration of the related Provider, but should still be validated
      3. Shipping
         * Country should default to Canada and not be updateable
           + Note: This should happen by default based on configuration of the related Provider, but should still be validated
      4. Payment
         * Update messaging at top of section to reference entering “Merchant ID and API Access Passcode” rather than “Merchant ID and Hash Key”
         * Add fields to configure and validate Bambora credentials
           + Merchant ID
           + API Access Passcode
           + “Test” button to validate credentials

See [Payment](#_Payment) section for more details on credential check

* + - * Remove Beanstream specific configuration fields, including those for CVD and AVS response code handling
        + Note: If these fields would not be shown to a Canada Partner unless Beanstream is selected as the Payment Gateway, then there would not be any code changes needed for this
      * Tax Registration fields for Canada should be removed as they are being moved to an earlier step (see above)
    1. Publish
       - Both English and French versions of the Company Website should be shown
         * Note: This functionality should already exist from the previous Canada implementation, but should still be validated
       - Display Bambora credentials
         * Merchant ID
       - Remove display of Beanstream specific configuration fields, including those for CVD and AVS response code handling
         * Note: If these fields would not be shown to a Canada Partner unless Beanstream is selected as the Payment Gateway, then there would not be any code changes needed for this
       - Display Canada Tax Registration fields
         * Note: This functionality should already exist from the previous Canada implementation, but should still be validated
    2. Confirmation
       - Both English and French version of the Supplies Assistant Download link and marketing collateral links should be shown
         * Note: This functionality should already exist from the previous Canada implementation, but should still be validated
  1. Pricing -> Price List
     1. Currency should default to Canadian Dollar and not be updateable for all pages where it is shown or used
        + Note: This should happen by default based on configuration of the related Provider, but should still be validated
  2. Pricing -> Shipping Rates
     1. Country should default to Canada and not be updateable for all pages where it is shown or used
        + Note: This should happen by default based on configuration of the related Provider, but should still be validated
  3. Customers > Customer & Revenue Overview -> Add Customer
     1. Country should default to Canada and not be updateable
        + Note: This should happen by default based on configuration of the related Provider, but should still be validated
     2. Ability to choose Customer language
        + English
        + French (Français)
        + Note: This functionality should already exist from the previous Canada implementation, but should still be validated, including not showing the language choice to US Partners
  4. Customers -> Customer & Revenue Overview -> Customer Details
     1. Add validation check to “Partner Ordering Enabled” field to not allow Partner to enable Partner Ordering for a customer unless they have “Payment Terms” of “Open Account”
        + Message = “You must change the Payment Terms for this customer to Open Account before enabling Partner Ordering.“
        + Note: Validation should occur for all VAR Partners (both US and Canada)
     2. Add validation check to “Payment Terms” field to not allow Partner to not allow Partner to change “Payment Terms” for a customer to “Credit Card” unless customer is not enabled for Partner ordering
        + Message = “You must disable Partner Ordering for this customer before changing their Payment Terms to Credit Card.“
        + Note: Validation should occur for all VAR Partners (both US and Canada)
     3. Canada specific Tax Class Codes should be shown (selectable)
        + Note: This functionality should already exist from the previous Canada implementation, but should still be validated
     4. Do not show (allow configuration of) an expiration date for a tax exemption
        + Note: This functionality should already exist from the previous Canada implementation, but should still be validated
  5. Customers -> Customer & Revenue Overview -> Customer Details -> Orders -> Order Detail
     1. Order details should include Canada specific display of sales tax amounts, tax exemption details, including certificate number, and Partner details, including tax registrations numbers (see [Tax](#_Tax) section for more details)
        + Note: This functionality should already exist from the previous Canada implementation, but should still be validated. Also, Partner details should be shown for both US and Canada orders.
  6. Customers -> Customer & Revenue Overview -> Customer Details -> Orders -> Shopping Cart
     1. Same requirement for displaying order details as above
        + Note: Partner details do not need to be shown on the Checkout page, but do need to be shown on the Order Submitted/Payment Problem page, while sales tax details need to be shown on both pages. Also, Partner details should be shown for both US and Canada orders.
  7. Orders -> Order List -> Order Detail
     1. Same requirement for displaying order details as above
  8. Preferences -> Account Details
     1. Ability to choose Partner language
        + English
        + French (Français)
        + Note: This functionality should already exist from the previous Canada implementation, but should still be validated, including not showing the language choice to US Partners
  9. Preferences -> System Settings -> Tax Configuration
     1. Add Tax Registration fields for Canada
        + GST/HST Registration ID
        + QST Registration ID
        + Note: Implementation of these fields should be the same as was previously done on the Tax Registration menu, including only requiring entry of “QST Registration ID” for Partners with a Quebec address
  10. Preferences -> System Settings -> eCommerce Account Configuration
      1. Rename tab to “Payment Methods” to align with US VAR implementation
      2. Add the ability to select/deselect “Open Account” and “Credit Card” payment methods
         + Note: Implementation of this should be the same as was previously done for US VAR Partners
      3. Add validation check to “Open Account” field to not allow Partner to disable “Open Account” payment method unless they have no customers enabled for Partner Ordering
         + Message = “You must disable Partner Ordering for all customers before removing Open Account as a Payment Method.“
         + Note: Validation should occur for all VAR Partners (both US and Canada)
      4. Add fields to configure and validate Bambora credentials
         + Merchant ID
         + API Access Passcode
         + “Test” button to validate credentials
           - See [Payment](#_Payment) section for more details on credentials check
      5. Update existing validation pop-up messaging shown when enabling Credit Card as a payment method as follows
         + Remove reference to “Authorize.net” from the message
         + Display support email address and support phone number as configured for the corresponding Program in the Provider Portal rather than hard-coding them in the message
         + Note: Above changes are intended to ensure that the same messaging can be used for both US and Canada Partners
      6. Remove Beanstream specific configuration fields, including those for CVD and AVS response code handling
         + Note: If these fields would not be shown to a Canada Partner unless Beanstream is selected as the Payment Gateway, then there would not be any code changes needed for this
  11. Preferences -> Messages
      1. Ability to preview messages in different languages
         + English
         + French (Français)
         + Note: This functionality should already exist from the previous Canada implementation, but should still be validated, including not showing the language choice to US Partners
  12. Preferences -> Branding
      1. Both English and French versions of the following should be shown
         + Company Website URL
         + Branded Customer Registration Page URL
         + Branded Supplies Assistant Desktop Application Download URL
         + Customer Portal URL
         + Note: This functionality should already exist from the previous Canada implementation, but should still be validated (the Customer Portal URL may not be shown since this is new since the last implementation)
  13. Preferences -> Users
      1. Ability to choose user language
         + English
         + French (Français)
         + Note: This functionality should already exist from the previous Canada implementation, but should still be validated, including not showing the language choice to US Partners
  14. Preferences -> Tax Registration
      1. Remove menu as content is being moved to the Tax Configuration menu (see above)
  15. Footer
      1. Remove “Sitemap” link
         + Note: This should be done for all Partners (both US and Canada)
  16. Other
      1. Do not display “Email Now” and “Chat Now” links

1. Questions
   1. Open
      1. None
   2. Closed
      1. Should the language choice that is made on the Login page update the user’s language preference in the DB?

**Answer:** Yes.

* + 1. Will the Chat feature be used for Canadian customers?

**Answer:** No. This feature will not be used for the initial implementation. This can be re-visited if it is decided to use it at some point in the future.

* + 1. Are there any additional configurations that need to be enabled to support the Payment provider that will be used in Canada?

**Answer:** No.

### Customer Portal

1. Language Localization
   1. All Customer facing content needs to be localized for French, including
      1. Tab titles
      2. Menu titles
      3. Column titles
      4. Field labels
      5. Button labels
      6. Link labels
      7. Selection choices in dropdown boxes
      8. Other static content, such as overview/explanatory text and error/warning messages
      9. Variable content that is localized via configuration in the Platform and Provider Portals
      10. Links and content that is opened or downloaded via those links, including
          * Learn about Rewards
          * Help
          * Provider Contact Information
            + Company website
      11. Footer links and content that is opened via the links
      12. Downloaded files and content
      13. All other Customer facing content not specifically mentioned above, with the exception of data entered directly by the Customer in the Customer Portal via input fields
   2. Language localization should include all eConcierge versions of the Customer Portal, including
      1. VAR
      2. NP
      3. NP+
   3. No localization of logos/branding is required
   4. Language localization approach should follow the same approach as used for the Customer Portal localization during the previous Canada implementation, with whatever updates/changes may be required due to the RWD implementation of the Customer Portal that was done since then
      1. Note: Any new Customer facing feature content that was not in place during the previous localization effort also needs to be included in the current effort
2. Currency Localization
   1. All financial details shown should be in Canadian dollars, using the same currency symbol and display format as currently used for US dollars
   2. Note: Above should happen by default due to the Provider being associated to Canada in the Platform Portal, so should not require any development work, unless changes are needed for this to work in the RWD portal
3. Address Localization
   1. All address details should be shown in a Canadian address format, including
      1. Use of “Postal Code” instead of “Zip Code”
      2. Use of “Province” instead of “State”
      3. Country shown as “Canada” (where applicable)
   2. Note: Above should happen by default due to the Provider being associated to Canada in the Platform Portal, so should not require any development work, unless changes are needed for this to work in the RWD portal
4. UI Changes
   1. Login Page
      1. Add the ability for user to select a language
         * English
         * French (Français)
      2. Language choice should be “sticky” across sessions
      3. Language choice should update the user’s language preference in the DB after log in (if different than the current preference)
         * Note: Language preference for a US Customer user should not be updated in the DB after login, even if it is different than their current preference. In other words, the language choice for a US Partner user as stored in the DB should only ever be English.
      4. Language choice should determine content shown to Customer on the login page and after log in (as listed above under Language Localization)
         * Note: Language choice should have no impact for a US Customer user after login. Their experience should be exactly the same as it currently is (everything shown in English), regardless of what language choice they may have made on the Login page.
   2. Self Registration Page
      1. Add the ability for user to select a language
         * English
         * French (Français)
         * Note: Language choice should not be shown on the registration page associated to US Partners
      2. Language choice should determine the following
         * Content shown to Customer during the registration process (as listed above under Language Localization)
         * Language preference persisted for the customer in the DB
   3. Account Menu -> Edit Users -> Primary User
      1. Add the ability for user to select a language
         * English
         * French (Français)
         * Note: Language choice should not be shown for US Customers
   4. Account Menu -> Edit Users -> Other Users -> Add/Edit User
      1. Add the ability to select a language for the user
         * English
         * French (Français)
         * Note: Language choice should not be shown for US Customers
   5. Orders -> Order Details
      1. Order details should include Canada specific display of sales tax amounts, tax exemption details, including certificate number, and Partner details, including tax registrations numbers (see [Tax](#_Tax) section for more details)
         * Note: Partner details should be shown for both US and Canada orders
   6. Cart -> Shopping Cart -> Payment
      1. Same requirement for displaying order details as above, with the exception that Partner details do not need to be shown, only sales tax details
      2. Display a new Payment section on left side of page (same area where existing Auth.net hosted payment form is displayed for US orders)
         * Section title = “Payment Information”
         * Credit Card section
           + Section title = “Credit Card Details”
           + Field 1

Field label = “Card Number”

Required

* + - * + Field 2

Field label = “Card Code”

Required

* + - * + Field 3

Field label = “Expiration Date”

Required

* + - * + Note: Credit Card entry fields will be hosted by Bambora using their Custom Checkout functionality and format and other validation of the fields will be handled using this functionality. See additional details in [Payment](#_Payment) section.
      * Billing Address section
        + Section title = “Billing Address”
        + Field 1

Field label = “First Name”

Free form text

Default initial value from customer primary contact

Required

Use standard name format validation

* + - * + Field 2

Field label = “Last Name”

Free form text

Default initial value from customer primary contact

Required

Use standard name format validation

* + - * + Field 3

Field label = “Address Line 1”

Free form text

Default initial value from customer company address

Required

Use standard address line format validation

* + - * + Field 4

Field label = “Address Line 2”

Free form text

Default initial value from customer company address

Optional

Use standard address line format validation

* + - * + Field 5

Field label = “City”

Free form text

Default initial value from customer company address

Required

Use standard city format validation

* + - * + Field 6

Field label = “Province”

Drop down box with listing of Canada Provinces

Default initial value from customer company address

Required

* + - * + Field 7

Field label = “Postal Code”

Free form text

Default initial value from customer company address

Required

Use standard postal code format validation

* + - * + Field 8

Field label = “Phone Number”

Free form text

Default initial value from customer primary contact

Required

Use standard phone number format validation

* + - * + ~~Field 9~~

~~Field label = “Email Address”~~

~~Free form text~~

~~Default initial value from customer primary contact~~

~~Required~~

~~Use standard email address format validation~~

Note: It was decided that the Email Address field would not be shown in the Payment section. However, it will still be passed to Bambora in the related API calls (See [Payment](#_Payment) section for additional details).

* + - * Button to proceed back to “Shipping” step
        + Button label = “Back to Shipping”
        + Positioned to left of “Place Order” button
        + Secondary styling
      * Button to initiate order placement process
        + Button label = “Place Order”
        + Positioned on lower right of page
        + Primary styling
        + Button should be disabled until all input fields have been entered and passed validation
        + See [Payment](#_Payment) section for additional details on order placement process
      * Other
        + Overall Payment section styling should follow the standard Customer Portal approach, with the intent that it fits in seamlessly from a look and feel perspective with the rest of the existing “Payment” page
        + Field labels should be displayed inside respective input fields and localized for French
        + Field validation messaging should be inline and localized for French
  1. Cart -> Shopping Cart -> Summary
     1. Same requirement for displaying order details as above, including showing both Partner details and sales tax details
        + Note: Partner details should be shown for both US and Canada orders
  2. Help
     1. Do not display “Customer Support Chat” field

1. Questions
   1. Open
      1. None
   2. Closed
      1. Should the language choice that is made on the Login page update the user’s language preference in the DB?

**Answer:** Yes.

* + 1. Will the Chat feature be used for Canadian customers?

**Answer:** No. This feature will not be used for the initial implementation. This can be re-visited if it is decided to use it at some point in the future.

### Supplies Assistant

1. Language Localization
   1. All Customer facing content needs to be localized for French, including
      1. Menu titles
      2. Column titles
      3. Field labels
      4. Button labels
      5. Link labels
      6. Other static content, such as overview/explanatory text, alert messages, and error/warning messages
      7. Variable content that is localized via configuration in the Platform and Provider Portals
      8. Links and content that is opened via the links, including
         * Learn about Rewards
         * Help
         * Contact
           + Company website
         * Account
         * Buy Online
      9. All other Customer facing content not specifically mentioned above, with the exception of data entered directly by the Customer in the Supplies Assistant via input fields
   2. Language localization should include the installation/setup wizard
   3. Language localization should include all eConcierge versions of the Supplies Assistant, including
      1. VAR
      2. NP
      3. NP+
   4. No localization of logos/branding is required
   5. Language localization approach should follow the same approach as used for the Supplies Assistant localization during the previous Canada implementation
      1. Note: Any new Customer facing feature content that was not in place during the previous localization effort also needs to be included in the current effort
2. Currency Localization
   1. No changes should be required as there are no financial details shown in the Supplies Assistant
3. Address Localization
   1. All address details should be shown in a Canadian address format, including
      1. Use of “Postal Code” instead of “Zip Code”
      2. Use of “Province” instead of “State”
   2. Note: This functionality should already exist from the previous Canada implementation, but should still be validated
4. UI Changes
   1. Ability for a customer to choose a language at time of installation should be re-enabled
      1. Note: Feasibility of defaulting this based on the language associated to the customer at time of creation should be investigated (this was not possible to during the previous Canada implementation)
   2. Ability for a customer to change their language preference from within the Supplies Assistant should be re-enabled
      1. Note: It is fine for a customer to need to log out and back in to the Supplies Assistant for this change to take effect
   3. Display of service loyalty icons and other service loyalty related details should occur for Canadian customers
      1. Note: This functionality was removed as part the previous Canada implementation, so this is being specifically called out in case there are code changes needed to cause them to display
5. Questions
   1. Open
      1. None
   2. Closed
      1. None

## Other Requirements

### Notifications

1. Appropriate Partner and Customer facing notification version (English or French) should be sent based on the chosen language of the notification recipient
   1. Note: This functionality should already exist from the previous Canada implementation, but should still be validated
2. Update existing “Order Details” merge tag that is used in the “ORDER PLACED” customer notification to include a billing information section for Open Account orders
   1. Details
      1. Section title = “Billing Information:”
         * Bold font
         * Positioned in same area as “Payment Information” section is for Credit Card orders
      2. Section Content
         * PO Number
           + Field Label = “Purchase Order #:”
           + Only include if customer has entered a PO number for the order
           + Field label and field content should be separated by a space
         * Customer First Name and Last Name
           + No field label needed
         * Company Name
           + No field label needed
         * Address Line 1
           + No field label needed
         * Address Line 2
           + Only include if customer has entered a second address line
           + No field label needed
         * City, State/Province and Zip/Postal Code
           + No field label needed
           + City and State/Province fields should be separated by a comma
           + State/Province and Zip/Postal Code should be separated by a space
         * Phone Number
           + Field label = “Phone:”
           + Field label and field content should be separated by a space
         * Source of data should be the same as is used for the “Billing” section content of the Customer Portal “Order Details” screen
         * Formatting of data should be the same as done for the “Shipping Address” section of the notification
      3. Other
         * Changes are only needed for orders with Open Account payment terms. Notification content can stay as-is for orders with Credit Card payment terms.
         * Change should be made for both US and Canada orders
3. Questions
   1. Open
      1. Are there any changes to notification content needed?

**Answer:** Discussion with the Canada tax people has indicated that Billing information should be shown on all customer invoices (so the “ORDER PLACED” notification), but it is currently only shown for orders with Credit Card payment terms, so updated above requirements for this. Other potential changes are still TBD.

* 1. Closed
     1. How will localized content in notification merge tags be handled?

**Answer:** This should be handled in the same way as done in the previous Canada implementation.

### Reports

1. TBD
2. Questions
   1. Open
      1. Are there any new or different report requirements?

**Answer:** It is currently assumed that no new reports will be needed for Canada, so this should primarily be based on what existing reports the Canada business decides are needed and then ensuring that those reports are properly localized.

* 1. Closed
     1. None

### Integration

#### **Distributor**

1. It is currently expected that the integration with Synnex Canada will be the same as what was done in the previous Canada implementation
2. Key integration differences to note (versus the Synnex US implementation)
   1. Different endpoint for posting requests (Stock Availability, Purchase Order, Purchase Order Status)
      1. Non-Prod
         * https://testec.synnex.ca/SynnexXML/PO
      2. Prod
         * https://ec.synnex.ca/SynnexXML/PO
   2. Different FTP site and credentials for retrieving ASN files
      1. Non-Prod
         * Server: 13.253.199.79
         * Username: synnex
         * Password: *Shared outside of this document*
         * Directory: /TEST\_SYNNEX\_CA
      2. Prod
         * Server: ftp.synnex.ca
         * Username: xeroxec
         * Password: *Shared outside of this document*
         * Directory: /
3. Questions
   1. Open
      1. None
   2. Closed
      1. Will any changes be required to the Synnex integration?

**Answer:** No. Synnex has confirmed that no changes will be needed.

* + 1. Will any new or different Synnex carrier codes need to be defined?

**Answer:** No. Synnex has confirmed that the list of carrier codes used for the original Canada implementation does not need to change.

#### **Tax**

1. Preliminary requirements for sales tax calculation and display will be the same as used in the previous Canada implementation
   1. [eConcierge Canada Requirements - Tax](http://xww.docushare.world.xerox.com/dsweb/Get/Document-5535058/eConcierge%20Canada%20Requirements%20-%20Tax%20Ver%201_6%20121022.docx)
2. Questions
   1. Open
      1. Have there been any changes to Canada sales tax requirements since the previous Canada implementation that will necessitate changes on our side?

**Answer:** TBD. Initial discussion with the Canada tax people has not highlighted anything needed beyond a change to the order invoice sent to the customer (see [Notifications](#_Notifications) section for more details), but these discussions are still on going.

* + 1. Will any new or different configuration of Vertex be required?

**Answer:** TBD. Note: Initial research indicates that there will be need to be some amount of reconfiguration of Vertex done (versus what was done for the original Canada implementation), but the exact details of this are still being determined.

* 1. Closed
     1. None

#### **Service**

1. It is currently expected that the integration with the OUT service system will be the same as what is done in the current US VAR implementation
2. Questions
   1. Open
      1. None
   2. Closed
      1. None

#### **Payment**

1. Integration Logic
   1. Custom Checkout
      1. Bambora Custom Checkout functionality should be integrated into the “Payment” page to allow for input and validation of the following fields
         * Card Number
         * Card Code
         * Expiration Date
      2. After customer clicks the “Place Order” button on the Customer Portal “Payment” page, a Bambora Custom Checkout “Create Token” call should be made to generate a token for the entered credit card details
         * Success Response
           + Defined as a scenario where a token is returned by the “Create Token” call
           + Create Auditlog entry for order indicating that the token creation succeeded
           + Proceed to Make Payment step
         * Failure Response
           + Defined as a scenario where an error is returned by the “Create Token” call
           + Create Auditlog entry for order indicating that the token creation failed
           + Customer should be shown a standard alert message

Appears over “Payment” page

Alert verbiage = “I'm sorry, we are currently experiencing system problems and are unable to process your order. Please try again later or contact Customer Support at ###-###-####.”

Support phone number should be the one configured for the corresponding Program in the Provider Portal

Button to close alert and show “Payment” page

Button label = “Close”

Positioned below alert verbiage

* 1. Make Payment
     1. After successful receipt of a credit card token from Bambora a Make Payment (Pre-Auth) request should be sent to Bambora using the token
        + Approved Response
          - Defined as a response returned by Bambora with an HTTP status of 200 and an “approved” value of 1
          - Proceed with next steps of the standard order processing flow for an approved Pre-Auth scenario, including creation of Order, Payment, and Auditlog records and sending of order to Synnex
          - DB Updates

Create Payment (X\_PAYMENT) record

ID = generated sequence number

Order ID = ID of order

Gateway ID = NULL

Response Code = “approved” value returned by Bambora

Response Reason Code = “message\_id” value returned by Bambora

Response Reason Text = “message” value returned by Bambora

Authorization Code = “auth\_code” value returned by Bambora

AVS Response Code = “id” value from “avs” variable returned by Bambora

Gateway Transaction ID = “id” value returned by Bambora

Amount = “amount” returned by Bambora

Payment Method = “payment\_method” value returned by Bambora

Transaction Type = “type” value returned by Bambora

Account Number = “last\_four” value returned by Bambora

Card Type = “card\_type” value returned by Bambora

First Name = “First Name” value entered on “Payment” page during order checkout

Last Name = “Last Name” entered on “Payment” page during order checkout

Address Line One = “Address Line 1” value entered on “Payment” page during order checkout

City Name = “City” value entered as entered on “Payment” page during order checkout

State Name = “Province” value entered on “Payment” page during order checkout

Postal Code = “Postal Code” value entered on “Payment” page during order checkout

Country Code = Static value of “CA”

Email = “Email” value from customer primary contact record

Phone Number = “Phone Number” value entered on “Payment’ page during order checkout

Finger Print Value = NULL

CVV Response Code = “cvd\_result” value returned by Bambora

Currency code = Static value of “CAD”

Customer ID = ID of customer associated to order

Who/when fields per standard approach

Note: Only Payment record creation is detailed above as all other DB updates occurring at this point in the process should follow the approach as currently used in the existing order processing flow

* + - * Declined Response
        + Defined as a response returned by Bambora that meets either of the following conditions

HTTP status of 200 and an “approved” value of 0

HTTP status of 402

* + - * + Order and Payment records should not be created and order should not be sent to Synnex
        + Create Auditlog entry for order indicating that the decline occurred
        + Customer should be shown a standard alert message

Appears over “Payment” page

Alert verbiage = “Your transaction could not be approved. Please try again.”

Button to close alert and show “Payment” page

Button label = “Close”

Positioned below alert verbiage

* + - * Error Response
        + Defined as a response returned by Bambora with an HTTP status other than 200 or 402
        + Order and Payment records should not be created and order should not be sent to Synnex
        + Create Auditlog entry for order indicating that the error occurred
        + Customer should be shown the existing “Payment Problem - Processing Error” page

Note: This should be the same page as used for the Auth.net integration if there are errors during the credit card processing

* 1. Complete Pre-Auth
     1. After successful confirmation of order by Synnex a Complete Pre-Auth request should be sent to Bambora using the Transaction ID received back on the Payment request. Note that in the case that Synnex does not confirm (rejects) the order it will be set to a cancelled state per the standard order processing flow for this scenario, but a void of the Pre-Auth will not be done like is done for US orders (as this void functionality is not supported in Canada). Instead, the pre-authorization will just expire on its own.
        + Approved Response
          - Defined as a response returned by Bambora with an HTTP status of 200 and an “approved” value of 1
          - Proceed with next step of the standard order processing flow for an approved Pre-Auth Completion scenario, including creation of Payment and Auditlog records
          - DB Updates

Create Payment (X\_PAYMENT) record

ID = generated sequence number

Order ID = ID of order

Gateway ID = NULL

Response Code = “approved” value returned by Bambora

Response Reason Code = “message\_id” value returned by Bambora

Response Reason Text = “message” value returned by Bambora

Authorization Code = “auth\_code” value returned by Bambora

AVS Response Code = “id” value from “avs” variable returned by Bambora

Gateway Transaction ID = “id” value returned by Bambora

Amount = “amount” returned by Bambora

Payment Method = “payment\_method” value returned by Bambora

Transaction Type = “type” value returned by Bambora

Account Number = “last\_four” value returned by Bambora

Card Type = “card\_type” value returned by Bambora

First Name = “First Name” value entered on “Payment” page during order checkout

Last Name = “Last Name” entered on “Payment” page during order checkout

Address Line One = “Address Line 1” value entered on “Payment” page during order checkout

City Name = “City” value entered as entered on “Payment” page during order checkout

State Name = “Province” value entered on “Payment” page during order checkout

Postal Code = “Postal Code” value entered on “Payment” page during order checkout

Country Code = Static value of “CA”

Email = “Email” value from customer primary contact record

Phone Number = “Phone Number” value entered on “Payment’ page during order checkout

Finger Print Value = NULL

CVV Response Code = “cvd\_result” value returned by Bambora

Currency code = Static value of “CAD”

Customer ID = ID of customer associated to order

Who/when fields per standard approach

Note: Only Payment record creation is detailed above as all other DB updates occurring at this point in the process should follow the approach as currently used in the existing order processing flow

* + - * Declined Response
        + Defined as a response returned by Bambora that meets either of the following conditions

HTTP status of 200 and an “approved” value of 0

HTTP status of 402

* + - * + Proceed with next step of the standard order processing flow for a declined Pre-Auth Completion scenario, including creation of Payment and Auditlog records and sending of the “SETTLEMENT ERROR“ notification to the related Partner

Note: In the case that the response returned by Bambora has an HTTP status of 402, the response body will have a different format and will not contain all of the necessary data elements to create the X\_PAYMENT record. So, for this scenario the X\_PAYMENT record will need to be created in a slightly different way (as detailed below).

* + - * + DB Updates

Create Payment (X\_PAYMENT) record (for HTTP status 200 decline scenario)

ID = generated sequence number

Order ID = ID of order

Gateway ID = NULL

Response Code = “approved” value returned by Bambora

Response Reason Code = “message\_id” value returned by Bambora

Response Reason Text = “message” value returned by Bambora

Authorization Code = “auth\_code” value returned by Bambora

AVS Response Code = “id” value from “avs” variable returned by Bambora

Gateway Transaction ID = “id” value returned by Bambora

Amount = “amount” returned by Bambora

Payment Method = “payment\_method” value returned by Bambora

Transaction Type = “type” value returned by Bambora

Account Number = “last\_four” value returned by Bambora

Card Type = “card\_type” value returned by Bambora

First Name = “First Name” value entered on “Payment” page during order checkout

Last Name = “Last Name” entered on “Payment” page during order checkout

Address Line One = “Address Line 1” value entered on “Payment” page during order checkout

City Name = “City” value entered as entered on “Payment” page during order checkout

State Name = “Province” value entered on “Payment” page during order checkout

Postal Code = “Postal Code” value entered on “Payment” page during order checkout

Country Code = Static value of “CA”

Email = “Email” value from customer primary contact record

Phone Number = “Phone Number” value entered on “Payment’ page during order checkout

Finger Print Value = NULL

CVV Response Code = “cvd\_result” value returned by Bambora

Currency code = Static value of “CAD”

Customer ID = ID of customer associated to order

Who/when fields per standard approach

Note: Only Payment record creation is detailed above as all other DB updates occurring at this point in the process should follow the approach as currently used in the existing order processing flow

Create Payment (X\_PAYMENT) record (for HTTP status 402 decline scenario)

ID = generated sequence number

Order ID = ID of order

Response Code = Static value of “0”

Response Reason Code = “code” value returned by Bambora

Response Reason Text = “message” value returned by Bambora

Authorization Code = NULL

AVS Response Code = NULL

Gateway Transaction ID = NULL

Amount = “Amount” value from corresponding Make Payment (Pre-Auth) Payment record

Payment Method = “Payment Method” value from corresponding Make Payment (Pre-Auth) Payment record

Transaction Type = Static value of “PAC”

Account Number = “Account Number” value from corresponding Make Payment (Pre-Auth) Payment record

Card Type = “Card Type” value from corresponding Make Payment (Pre-Auth) Payment record

First Name = “First Name” value entered on “Payment” page during order checkout

Last Name = “Last Name” entered on “Payment” page during order checkout

Address Line One = “Address Line 1” value entered on “Payment” page during order checkout

City Name = “City” value entered as entered on “Payment” page during order checkout

State Name = “Province” value entered on “Payment” page during order checkout

Postal Code = “Postal Code” value entered on “Payment” page during order checkout

Country Code = Static value of “CA”

Email = “Email” value from customer primary contact record

Phone Number = “Phone Number” value entered on “Payment’ page during order checkout

Finger Print Value = NULL

CVV Response Code = NULL

Currency code = Static value of “CAD”

Customer ID = ID of customer associated to order

Who/when fields per standard approach

Note: Only Payment record creation is detailed above as all other DB updates occurring at this point in the process should follow the approach as currently used in the existing order processing flow

* + - * Error Response
        + Defined as a response returned by Bambora with an HTTP status other than 200 or 402
        + Proceed with next step of the standard order processing flow for a failed Pre-Auth Completion scenario, including creation of an Auditlog record and sending of the “SETTLEMENT ERROR“ notification to Partner
  1. Get Payment
     1. After Partner clicks the “Test” button for Bambara credentials on the Partner Portal “Payment Methods” page, a Get Payment request should be sent to Bambora using a dummy Transaction ID
     2. Valid Credentials Response
        + Defined as a response with an HTTP status of 200 or 404
          - HTTP status codes for valid Bambora credentials should be configurable in the DB (X\_TB\_CONFIGURATION table) rather than hard-coded to allow them to be easily updated in the future as needed. Initial values for the configuration should be the two codes mentioned above.
        + Display standard valid credentials pop-up messaging (same as currently used for Auth.net integration)
     3. Invalid Credentials Response
        + Defined as a response with an HTTP status of 400 or 401
          - HTTP status codes for invalid valid Bambora credentials should be configurable in the DB (X\_TB\_CONFIGURATION table) rather than hard-coded to allow them to be easily updated in the future as needed. Initial values for the configuration should be the two codes mentioned above.
        + Display standard invalid credentials pop-up messaging (same as currently used for Auth.net integration)
     4. Other or Unknown Response
        + Defined as a response with an HTTP status other than those used to determine whether the credentials are valid or invalid
          - Logic for this should be that the HTTP status code is not contained in the list of Bambora valid and include credential status codes as configured in the DB (X\_TB\_CONFIGURATION table) rather than hard-coding the status codes.
        + Display standard error/system not responding pop-up messaging (same as currently used for Auth.net integration)

1. Technical Details
   1. Custom Checkout
      1. Documentation
         * <https://dev.na.bambora.com/docs/guides/custom_checkout/>
      2. Details
         * Bambora Custom Checkout JavaScript should be loaded into “Payment” page and initialized
           + https://libs.na.bambora.com/customcheckout/1/customcheckout.js
         * Credit card input fields should be created with appropriate labels and styling and mounted in the “Payment” page using the “create” and “mount” functions
           + Labels and styling can be passed to the “create” functions using the “options” argument
         * Event listeners should be used for the credit card input fields to handle input validation
         * A one-time use token should be created for the credit card details using the “createToken” function
         * Refer to the Bambora Custom Checkout documentation and code samples for more details
   2. Make Payment
      1. TB -> Bambora via REST API
      2. Configuration
         * API details should be configured in DB (X\_API\_CONFIG and X\_API\_CONFIG\_DETAIL)
      3. Documentation
         * <https://dev.na.bambora.com/docs/references/payment_APIs/v1-0-5/>
      4. Request Details
         * Endpoint
           + https://api.na.bambora.com/v1/payments
           + Same endpoint will be used for all environments (Prod and Non-Prod)
         * Method
           + POST
         * Header
           + Details

|  |  |
| --- | --- |
| Key | Value |
| Authorization | Static value of “Passcode” followed by a Base64 encoded value of the Partner’s “Merchant ID” and “API Access Passcode” values entered in the Partner Portal separated by a colon(:) (i.e. encoded value of Merchant ID:API Access Passcode) |
| Content-Type | Static value of “application/json” |

* + - * + Sample

|  |
| --- |
| Authorization: Passcode MzAwMjA3MDM0OkFFMTBENTBFMTA2MzQxZDRhMTQCV  Content-Type: application/json |

* + - * Body
        + Format

JSON

* + - * + Details

|  |  |  |
| --- | --- | --- |
| Name | Value | Type |
| order\_number | eConcierge order number | String |
| amount | Total amount of order | Number |
| payment\_method | Static value of “token” | String |
| name | “First Name” and “Last Name” values (separated by a space) entered on “Payment” page during order checkout. Part of “billing” variable. | String |
| address\_line1 | “Address Line 1” value entered on “Payment” page during order checkout. Part of “billing” variable. | String |
| address\_line2 | “Address Line 2” value entered on “Payment” page during order checkout. Part of “billing” variable. | String |
| city | “City” entered on “Payment” page during order checkout. Part of “billing” variable. | String |
| province | 2 character “Province” code value entered on “Payment” page during order checkout. Part of “billing” variable. | String |
| country | Static value of “CA”. Part of “billing” variable. | String |
| phone\_number | “Phone Number” value entered on “Payment” page during order checkout. Part of “billing” variable. | String |
| email\_address | “Email” value from customer primary contact record.  Part of “billing” variable. | String |
| complete | Static value of false. Part of “token” variable. | Boolean |
| code | “Token” value returned by Bambora during “createToken” call. Part of “token” variable. | String |
| name | “First Name” and “Last Name” values (separated by a space) entered on “Payment” page during order checkout. Part of “token” variable. | String |

* + - * + Sample

|  |
| --- |
| {  "order\_number":"EC-12345",  "amount":100.00,  "payment\_method":"token",  "billing": {  "name":"Joe Smith",  "address\_line1":"211 Bear St",  "address\_line2":"Suite 123",  "city":"Banff",  "province":"AB",  "country":"CA",  "postal\_code":"T1L 1A8",  "phone\_number":"6047112356",  "email\_address":"joe.smith@test.com"  },  "token": {  "complete":false,  "code":"a64-5db5dbe7-77e2-4456-b25b-11b8007ab3fc",  "name":"Joe Smith"  }  } |

* + - * Other
        + Full request details should be captured in DB (X\_API\_REQUEST)

Note: This should include the request URL, request header, and request body (as applicable for the request)

* + 1. Response Details
       - Status

|  |  |
| --- | --- |
| Code | Description |
| 200 | Payment response containing the payment details as well as if the payment was approved or declined. |
| 400 | Bad Request |
| 401 | Authentication Failure |
| 402 | Business Rule Violation or Decline |
| 403 | Authorization Failure |
| 404 | Not Found |
| 405 | Invalid Request Method |
| 500 | Internal Server Error |

* + - * Header
        + Response headers will vary dependent on status of response, but there should not be any values in the header that are needed for processing the response
      * Body
        + Format

JSON

* + - * + Details

|  |  |  |
| --- | --- | --- |
| Name | Value | Type |
| id | Transaction ID | String |
| approved | Approval status of transaction. 0 for declined, 1 for approved | Integer |
| message\_id | Payment response code | Integer |
| message | Message containing information about status of transaction | String |
| auth\_code | Authorization code of transaction | String |
| type | Payment transaction type. Should be “PA”. | String |
| payment\_method | Payment method type. Should be “CC”. | String |
| amount | Amount of transaction | Number |
| card\_type | Credit card type. Part of “card” variable. | String |
| last\_four | Last four digit of credit card number. Part of “card” variable. | String |
| cvd\_result | CVD verification result. Part of “card” variable. | String |
| id | AVS verification result. Part of “avs” variable. | String |

Note: Values not mentioned should not be needed as part of response processing logic.

* + - * + Sample

|  |
| --- |
| {  "id": "10000020",  "authorizing\_merchant\_id": 300207034,  "approved": "1",  "message\_id": "1",  "message": "Approved",  "auth\_code": "TEST",  "created": "2019-08-07T13:13:46",  "order\_number": "EC-12345",  "type": "PA",  "payment\_method": "CC",  "risk\_score": 0.0,  "amount": 100.00,  "custom": {  "ref1": "",  "ref2": "",  "ref3": "",  "ref4": "",  "ref5": ""  },  "card": {  "card\_type": "VI",  "last\_four": "1234",  "address\_match": 0,  "postal\_result": 0,  "avs\_result": "0",  "cvd\_result": "1",  "avs": {  "id": "N",  "message": "Street address and Postal/ZIP do not match.",  "processed": true  }  },  "links": [  {  "rel": "complete",  "href": "https://api.na.bambora.com/v1/payments/10000020/completions",  "method": "POST"  }  ]  } |

* + - * Other
        + Full response details, including HTTPS status, should be captured in DB (X\_API\_RESPONSE)
  1. Complete Pre-Auth
     1. TB -> Bambora via REST API
     2. Configuration
        + API details should be configured in DB (X\_API\_CONFIG and X\_API\_CONFIG\_DETAIL)
     3. Documentation
        + <https://dev.na.bambora.com/docs/references/payment_APIs/v1-0-5/>
     4. Request Details
        + Endpoint
          - https://api.na.bambora.com/v1/payments/{transId}/completions
          - Same endpoint will be used for all environments (Prod and Non-Prod)
        + Method
          - POST
        + Header
          - Details

|  |  |
| --- | --- |
| Key | Value |
| Authorization | Static value of “Passcode” followed by a Base64 encoded value of the Partner’s “Merchant ID” and “API Access Passcode” values entered in the Partner Portal separated by a colon(:) (i.e. encoded value of Merchant ID:API Access Passcode) |
| Content-Type | Static value of “application/json” |

* + - * + Sample

|  |
| --- |
| Authorization: Passcode MzAwMjA3MDM0OkFFMTBENTBFMTA2MzQxZDRhMTQCV  Content-Type: application/json |

* + - * Body
        + Format

JSON

* + - * + Details

|  |  |  |
| --- | --- | --- |
| Name | Value | Type |
| order\_number | eConcierge order number | String |
| amount | Total amount of order | Number |

* + - * + Sample

|  |
| --- |
| {  "order\_number":"EC-12345",  "amount":100.00  } |

* + - * Other
        + The transId value sent as part of the request URL should be the “id” value returned by Bambara on the corresponding Make Payment (Pre-Auth) response
        + Full request details should be captured in DB (X\_API\_REQUEST)

Note: This should include the request URL, request header, and request body (as applicable for the request)

* + 1. Response Details
       - Status

|  |  |
| --- | --- |
| Code | Description |
| 200 | Payment response containing the payment details as well as if the payment was approved or declined. |
| 400 | Bad Request |
| 401 | Authentication Failure |
| 402 | Business Rule Violation or Decline |
| 403 | Authorization Failure |
| 404 | Not Found |
| 405 | Invalid Request Method |
| 500 | Internal Server Error |

* + - * Header
        + Response headers will vary dependent on status of response, but there should not be any values in the header that are needed for processing the response
      * Body
        + Format

JSON

* + - * + Details

|  |  |  |
| --- | --- | --- |
| Name | Value | Type |
| id | Transaction ID | String |
| approved | Approval status of transaction. 0 for declined, 1 for approved | Integer |
| message\_id | Payment response code | Integer |
| message | Message containing information about status of transaction | String |
| auth\_code | Authorization code of transaction | String |
| type | Payment transaction type. Should be “PA”. | String |
| payment\_method | Payment method type. Should be “CC”. | String |
| amount | Amount of transaction | Number |
| card\_type | Credit card type. Part of “card” variable. | String |
| last\_four | Last four digit of credit card number. Part of “card” variable. | String |
| cvd\_result | CVD verification result. Part of “card” variable. | String |
| id | AVS verification result. Part of “avs” variable. | String |

Note: Values not mentioned should not be needed as part of response processing logic.

* + - * + Sample

|  |
| --- |
| {  "id": "10000022",  "authorizing\_merchant\_id": 300207034,  "approved": "1",  "message\_id": "1",  "message": "Approved",  "auth\_code": "TEST",  "created": "2019-08-08T10:37:56",  "order\_number": "EC-54321",  "type": "PAC",  "payment\_method": "CC",  "risk\_score": 0.0,  "amount": 100.00,  "custom": {  "ref1": "",  "ref2": "",  "ref3": "",  "ref4": "",  "ref5": ""  },  "card": {  "card\_type": "VI",  "address\_match": 0,  "postal\_result": 0,  "avs\_result": "0",  "cvd\_result": "1",  "cavv\_result": " ",  "avs": {  "id": "U",  "message": "Address information is unavailable.",  "processed": false  }  },  "links": [  {  "rel": "return",  "href": "https://api.na.bambora.com/v1/payments/10000022/returns",  "method": "POST"  },  {  "rel": "complete",  "href": "https://api.na.bambora.com/v1/payments/10000022/completions",  "method": "POST"  }  ]  } |

* + - * Other
        + Full response details, including HTTPS status, should be captured in DB (X\_API\_RESPONSE)
  1. Get Payment
     1. TB -> Bambora via REST API
     2. Configuration
        + API details should be configured in DB (X\_API\_CONFIG and X\_API\_CONFIG\_DETAIL)
     3. Documentation
        + <https://dev.na.bambora.com/docs/references/payment_APIs/v1-0-5/>
     4. Request Details
        + Endpoint
          - https://api.na.bambora.com/v1/payments/{transId}
          - Same endpoint will be used for all environments (Prod and Non-Prod)
        + Method
          - GET
        + Header
          - Details

|  |  |
| --- | --- |
| Key | Value |
| Authorization | Static value of “Passcode” followed by a Base64 encoded value of the Partner’s “Merchant ID” and “API Access Passcode” values entered in the Partner Portal separated by a colon(:) (i.e. encoded value of Merchant ID:API Access Passcode) |
| Content-Type | Static value of “application/json” |

* + - * + Sample

|  |
| --- |
| Authorization: Passcode MzAwMjA3MDM0OkFFMTBENTBFMTA2MzQxZDRhMTQCV  Content-Type: application/json |

* + - * Body
        + None
      * Other
        + The transId value sent as part of the request URL should be a static dummy value of 1
        + Full request details should be captured in DB (X\_API\_REQUEST)

Note: This should include the request URL, request header, and request body (as applicable for the request)

* + 1. Response Details
       - Status

|  |  |
| --- | --- |
| Code | Description |
| 200 | Transaction object |
| 400 | Bad Request |
| 401 | Authentication Failure |
| 402 | Business Rule Violation or Decline |
| 403 | Authorization Failure |
| 404 | Not Found |
| 405 | Invalid Request Method |
| 500 | Internal Server Error |

* + - * Header
        + Response headers will vary dependent on status of response, but there should not be any values in the header that are needed for processing the response
      * Body
        + Body content will vary dependent on status of response, but there should not be any values in the body that are needed for processing the response
      * Other
        + Full response details, including HTTPS status, should be captured in DB (X\_API\_RESPONSE)

1. Questions
   1. Open
      1. None
   2. Closed
      1. Should non-Canada billing addresses be allowed in the Payment page during order checkout?

**Answer:** No. It was decided to not allow this at this point in time as it would add additional complexity to the Payment page for what seems like would be a very rare use case.

* + 1. What is the best approach for handling of AVS and CVD failures?

**Answer:** After further internal discussion as well as feedback from the Bambara team it was decided that we would not implement any special or custom handling of AVS and CVD failures as was done in the previous Canada implementation. This would add a lot of complexity and potential for poor customer experience to the implementation with minimal added value in regards to fraud prevention.

* + 1. Is there additional response handling logic needed to differentiate between a decline and error scenario on the Make Payment (Pre-Auth) request and Complete Pre-Auth request.

**Answer:** Yes. Further discussion with Bambora has indicated that responses returned with an HTTP status of 402 should also be treated as a decline rather than as an error. The above requirements have been updated to reflect this.

## Functional Spec

N/A

## Wireframes

* Update existing Platform Portal, Provider Portal, and Partner Portal Wireframe documents as needed
* Update existing Customer Portal VBS documents as needed
* Update existing Supplies Assistant VBS documents as needed

## Approach Document

Development of new technical approach document(s) will be required for all impacted areas

## Configuration

See details in requirement sections above

## Open Questions

See details in requirement sections above