entryLINK System Integration Guide

Last Updated - October 2026

This document outlines how third-party applications can electronically interchange data with the ProVia® entryLINK system.

It provides a developer with the needed resources to automate the sign-in process and to create, retrieve and update order information.

For support, please contact ProVia entryLINK Support at 800-669-4711.

Getting Started

All entryLINK integration is web based and driven by GET requests to the following URL:
https://entrylink.provia.com/integrate.aspx

All requests must include the following query string parameters:

- m - mode
- u - user name

Valid values for the m (mode) variable are:

- login
- getorders
- getorder2
- createorder
- createorderjson
- setcustomer
- getlocations
- setlocation
- setpo
- setjob
- additem
- copyline
- setcomment
- settag
- submitorder
- getpic
- shipmanifest
- rmalist
- ofpm
- delitem
- delord
- chgqty
- editunit

Deprecated Modes:

- getorder

Supply the password

The user's password can be supplied either as a "p" query string parameter (as shown in the links below, deprecated) or via a request header named "p" (preferred).

If the username or password is not valid, the system will return a 401-Unauthorized error.

Mode login

The “login” mode is used to automatically log a user into the system, bypassing the login screen. You can use an “o” parameter to specify an entryLINK order number, and the user will be taken directly to that order. If the “o” parameter is not specified, the user will be taken to the default index page.

You could use this feature to create a “Show ProVia Order” button inside your application. When the user clicks this button they could be taken directly to the order.

The parameter “SingleQuote=on” will lock the user into only viewing a single quote.

Example URL:

https://entrylink.provia.com/integrate.aspx?m=login&u=demo&p=ProVia01&o=1234567

Portal Logins

If this is a legacy/deprecated “portal” login (which requires the “Portal Login” slider to be activated when logging in) add “&ut=2” to the integration URL.

Portal logins will be 8-digit numerics.

Mode getorders

The “getorders” mode returns an XML or JSON document of open orders in the system. Orders with the following criteria will be included:

- All open orders and quotes entered in the past 30 days.
- All orders currently in progress.
- All orders invoiced in the past 30 days.

For each order, the following fields are included:

- Order Number
- PO Number
- Job Name
- Customer Number
- Customer Location Name
- Status
- Order Date
- Receive Date
- Ship Date
- Sub Total
- Tax
- Total

You could use this feature to get an open order listing, to find the ProVia order number for your PO Number, or to import/synchronize ship dates, etc.

Example URL:

https://entrylink.provia.com/integrate.aspx?m=getorders&u=demo&p=ProVia01

You can limit the returned results using the following parameters:

- po – filters the PO Number (uses like *search* matching)
- job – filters the Job Name (uses like *search* matching)

The default response is XML, however “&fmt=json” can be added to the URL to return JSON.

Mode getorder2

The “getorder2” mode returns an XML or JSON document describing and documenting a specific order or quote in the system.

This is the latest version.

You can use this to import full order details into your system.

Use the “o” parameter to specify the entryLINK order number.

Example URL:

https://entrylink.provia.com/integrate.aspx?m=getorder2&u=demo&p=ProVia01&o=1234567

The default response is XML, however “&fmt=json” can be added to the URL to return JSON.

Mode getorder

This mode is deprecated; it had inconsistencies in how pricing was presented, and did not support price details for door and window products.

If your application is still using getorder, we encourage you to switch to using getorder2.

Mode createorder

The "createorder" mode creates a new order in the entryLINK system. The entryLINK order number is returned. The following optional query string parameters can be added:

- po – Your PO Number
- job – Your Job Name
- branch – the ship-to location. Must be a valid ProVia customer number.

Example URL:

https://entrylink.provia.com/integrate.aspx?m=createorder&u=demo&p=ProVia01&po=99754&job=SMITH

Mode createorderjson

The "createorderjson" mode allows creating new orders in the entryLINK system by POSTing a JSON payload. A JSON response is returned, containing the order number and a Status (Ok, or an error statement).

You can add "&submit=on" to the query string, which will cause the order to submit if it is error free.

Example URL:

https://entrylink.provia.com/integrate.aspx?m=createorderjson&u=demo&p=ppp

Example JSON post body:

```
{
 "CustNo":"999001",
 "PONumber":"AAAA",
 "JobName":"BBBB",
 "Lines":[ {"PartNo":"H-SI-45241-19","Qty":6},
        {"PartNo":"H-SI-45241-10","Qty":12}
       ]
}
```

Notes

The customer number, PO Number and Job Name are all optional.

The customer must be a valid ProVia ship-to customer number that is visible to that customer login.

Mode setcustomer

The “setcustomer” mode allows you to set the customer information on a specific order. You can either use the portal ID to link the order to an existing/repeat customer, or set each field individually for a one-time customer.

Note – this is setting the homeowner or contractor information NOT the ship-to location.

The following query string parameters are accepted:

- o – entryLINK Order Number (required)
- pid – portal id (if this is specified, the following parameters are not valid and will be ignored.)
- name – the customer’s name
- add1 – customer’s address line 1
- add2 – customer’s address line 2
- city – customer’s city
- state – customer’s state
- zip – customer’s zip-code
- phone – customer’s phone #
- fax – customer’s fax
- email – customer’s email address

Example URL:

https://entrylink.provia.com/integrate.aspx?m=setcustomer&u=demo&p=ProVia01&o=1234567&name=Jim+Smith&city=Chicago

Mode getlocations

The “getlocations” mode provides a JSON response that lists available ship-to customer locations visible to the user. This allows you to set the ship-to location using the setlocation or createorder modes.

Mode setlocation

The “setlocation” mode allows you to specify the ship-to location for the order.

The following query string parameters are required:

- o – entryLINK Order Number
- cn – The ship-to customer location

The system will return “ok” or error text – or use “&fmt=json” for a proper JSON response.

Mode setpo

The "setpo" mode allows you to set the PO Number field after an order has been created and is in "Unsubmitted" status. The field length is 20 characters, and the value is forced to uppercase. The API will return an indication if the PO number duplicates that of another order.

The following query string parameters are accepted:

- o – entryLINK Order Number (required)
- po – Your PO Number

Example URL:

https://entrylink.provia.com/integrate.aspx?m=setpo&u=demo&p=ProVia01&o=1234567&po=AAAAAA

The system will return "ok" or error text – or use "&fmt=json" for a proper JSON response.

Mode setjob

The "setjob" mode allows you to set the Job Name field after an order has been created and is in "Unsubmitted" status. The field length is 20 characters, and the value is forced to uppercase.

The following query string parameters are accepted:

- o – entryLINK Order Number (required)
- job – Your Job Name

Example URL:

https://entrylink.provia.com/integrate.aspx?m=setjob&u=demo&p=ProVia01&o=1234567&job=SMITH

The system will return "ok" or error text – or use "&fmt=json" for a proper JSON response.

Mode additem

The "additem" mode allows you to add an item to the order. You must specify the ProVia item number and a quantity. Contact customer service for an item number listing.

NOTE – this feature is most useful for ordering marketing materials or siding and stone products. Ordering configured products (doors and windows) is NOT supported via this method.

The following query string parameters are required:

- o – Order Number
- i – Item Number
- q – Quantity

To initialize Door/Window products, use the following codes:
ED-Entry Door, VW-Vinyl Window, VP-Vinyl Patio, SD-Storm Door

The system will return "ok" or error text – or use "&fmt=json" for a proper JSON response.

Example URL:

https://entrylink.provia.com/integrate.aspx?m=additem&u=demo&p=ProVia01&o=1234567&i=xxxxx&q=1

Configured Items (Deprecated; use EditUnit)

Configured items can be uploaded by POST-ing the property bag (a list of key/value pairs). Place each key/value pair on its own line, and separate them with a comma – such as:

Color; white

Width; 36

Height; 80

Contact ProVia for a listing/visibility of the property bag details.

Mode copyline

The "copyline" mode allows you to copy an existing item on an order.

The following query string parameters are supported:

- o – Destination Order Number (required)
- fo – Source Order Number (optional – if omitted destination order is assumed)
- l – Source Line Number (required)

The system will return "ok" or error text – or use "&fmt=json" for a proper JSON response.

Mode setcomment

The "setcomment" mode allows you to specify line comments for a line on the order.

The following query string parameters are required:

- o – Order Number
- l – Line Number
- The comment information must be POST-ed

The system will return "ok" or error text – or use "&fmt=json" for a proper JSON response.

Mode settag

The "settag" mode allows you to specify a line tag for a line on the order (such as "front door" or "bathroom window").

The following query string parameters are required:

- o – Order Number
- l – Line Number
- The tag name must be POST-ed (35 characters max)

The system will return "ok" or error text – or use "&fmt=json" for a proper JSON response.

Mode submitorder

The "submitorder" mode allows you to submit the order to ProVia for production. After the order is submitted you will no longer be able to edit it.

NOTE – the order may be in a state where it cannot be submitted. For example, if the order is empty or has a duplicate PO number. In these cases, the API will return error text. If everything is ok it will return blank and the order will be submitted.

The following query string parameters are required:

- o – Order Number

Example URL:

https://entrylink.provia.com/integrate.aspx?m=submitorder&u=demo&p=ProVia01&o=1234567

The system will return "ok" or error text – or use "&fmt=json" for a proper JSON response.

Mode getpic

The "getpic" mode returns a URL to a product image.

The following query string parameters are required:

- o – Order Number
- l – Line Number
- v – view direction – either "i" inside or "o" outside (optional)

Example URL:

https://entrylink.provia.com/integrate.aspx?m=getpic&o=8148936&l=1&v=i&u=Demo&p=pppp

Mode shipmanifest

The "shipmanifest" mode allows you to acquire a shipping manifest of recently shipped orders. In addition, you can add a "fmt" element to the querystring to specify the output format. Valid values are CSV, JSON, and XML.

For specific date ranges; "sd" and "ed" can be added to box dates.

Example URLs:

https://entrylink.provia.com/integrate.aspx?m=shipmanifest&u=demo&p=ProVia01&fmt=CSV

https://entrylink.provia.com/integrate.aspx?m=shipmanifest&u=demo&p=ProVia01&fmt=JSON

https://entrylink.provia.com/integrate.aspx?m=shipmanifest&u=demo&p=ProVia01&fmt=XML

Example with dates:

https://entrylink.provia.com/integrate.aspx?m=shipmanifest&u=demo&p=ProVia01&fmt=CSV&sd=1/1/2026&ed=1/10/2026

Mode RMAList

The “rmalist” mode allows you to acquire a listing of active RMAs (Return Merchandise Authorizations). You can add a “fmt” element to the querystring to specify the output format. Valid values are CSV, JSON, and XML.

Example URLs:

https://entrylink.provia.com/integrate.aspx?m=rmalist&u=demo&p=ProVia01&fmt=CSV

https://entrylink.provia.com/integrate.aspx?m=rmalist&u=demo&p=ProVia01&fmt=JSON

https://entrylink.provia.com/integrate.aspx?m=rmalist&u=demo&p=ProVia01&fmt=XML

Mode OFPM

The “OFPM” mode (Order Fulfillment Prediction Model) allows you to acquire the output from ProVia’s Order Fulfillment Prediction Model, providing guidance on ProVia’s expected delivery dates by product line.

You must provide the “cn” parameter, specifying your ProVia ship-to customer number, as the truck routes and drop-dates vary by location.

The output currently is only CSV.

Example URL:

https://entrylink.provia.com/integrate.aspx?m=ofpm&u=demo&p=ProVia01&cn=999001

Mode delitem

The “delitem” mode allows you to delete an item on an order. The following query string parameters are required:

- o – Order Number
- l – Line Number

The system will return “ok” or error text – or use “&fmt=json” for a proper JSON response.

Mode delord

The “delord” mode allows you to delete an order.

The following query string parameters are required:

- o – Order Number

The system will return “ok” or error text – or use “&fmt=json” for a proper JSON response.

Mode chgqty

The “chgqty” mode allows you to change the quantity of an item on an order. You must specify the ProVia line number and the quantity.

The following query string parameters are required:

- o – Order Number
- l – Line Number
- q – New Order Quantity

The system will return “ok” or error text – or use “&fmt=json” for a proper JSON response.

Mode editunit

The “editunit” mode provides a full API that allows third-party developers to create a robust door/window editing application that integrates tightly with ProVia’s entryLINK system.

An application API Key is required to access this API. Initially the application API Key will allow editing, but not saving, units. Contact ProVia to request an API key.
(This API key is specific to the developed application, not the customer.)

To access the API initially you will need to supply:

- o – Order Number
- l – line number
- apikey – the provided API key (via headers, not URL)

The system will respond with a JSON document, including:

- Status (string) – monitor for errors or issues, or blank if ok
- StatusOk (bool) - monitor for errors or issues
- ProductDescription (html string) – product description to be presented to the user
- Sizing (html string) – product sizing to be presented to the user
- InsideImageURL – product visual (inside looking out) to be presented to the user
- OutsideImageURL – product visual (outside looking in) to be presented to the user
- Tabs – a key/value pair (int/string) of available user interface tabs, such as “Style”, “Color” etc.
- Options – a list of User Interface options, including the option names, values, selections, images, etc.
- Warnings – a list of Warnings to present to the user (with multiple types, information, warning, critical, question, click-to-acknowledge, etc)
- State (string) – a state variable that must be passed back.
- PriceDetails – a list of detailed price information, contains list, cost and sell depending on the user preferences.
- PriceList (decimal) – the list price (in qty 1)
- PriceCost (decimal) – unit cost (in qty 1)
- PriceSell (decimal) – unit sell price (in qty 1)
- PriceListExt (decimal) – the list price (x qty)
- PriceCostExt (decimal) – unit cost (x qty)
- PriceSellExt (decimal) – unit sell price (x qty)
  Note – price visibility varies by user permission and preference

  Window Properties – see the “Window” section below:
- WindowRenderPostions
- WindowSelectAll
- WindowCurrentCell
- WindowOverlayMode

Edit Unit Workflow –

Initial request: Call the “editunit” URL and retrieve and parse the JSON.

Inspect the Status / StatusOK fields –

They could be something like “Unit Not Editable” / False. In this case message the user and return to your application.

If state is OK, you can proceed to construct the editing user interface –

- Present the product description, with a toggle to present Sizing or Pricing instead.
- Present the product visual; with a toggle to switch between inside and outside view.
- Present the list of tabs – this will include things like “Style”, “Color”, “Glass” etc.
- Present the list of warnings; this can include a list of questions that still need to be answered, product information, or “click to acknowledge” warnings.
- Present the list of options.

Each option will have a “Name” associated with it, and each selection will have a value (for example Color and 1=White, 2=Green, 3=Black).

To make an option selection and cycle the U/I - call the same “editunit” URL but include

- k – the selected option name (“color”)
- v – the selected option value (“3” for black)
  i.e.: &k=color&v=3

POST the “State” variable back as the body of the request

Note the first request is GET, but all subsequent requests are POST.

The response will cycle with the change applied and you can refresh the user interface and continue.

To change tabs (switch from the Style to Glass tab):
&k=tab&v=3 (the number associated with the tab)

To acknowledge a warning:

`&k=ack_wrn:keyword&v=on`

Windows

Mulled windows have several features that require special handling. Specifically, selecting which window in the mulled set is "in focus" for option selection.

To facilitate this, this API includes several window-specific features including:

The **WindowOverlayMode** property – which controls whether overlays are needed:

0 – None
1 – Show overlays with NO Select All button
2 – Show overlays WITH a Select All button

The **WindowRenderPositions** property which is a list of WindowRenderPosition.

Each WindowRenderPosition includes a CellID (such as A1, B2, etc), a width, height and a left and top position. These coordinates map to the outside view of the rendered product visualization. Selectors must be scaled and applied over the rendering, allowing the user to "touch to select" a specific window.

When the user performs this action, send over the selected cell as: &k=last_cell&v=B2

The **WindowCurrentCell** property which indicates which Cell ID should be highlighted.

The **WindowSelectAll** property, which indicates if window "select all" is ON or OFF.
The WindowCurrentCell fields is only applicable when this is False.
When the user toggles this selector, send over the selection with:

&k=cell_sel_all&v=true OR &k=cell_sel_all&v=true

Save / Cancel

The user interface should provide "Save" and "Cancel" buttons.

If the user clicks "Cancel" simply return to the prior screen in your application.

To "Save" use: &k=save&v=on – this will write the changes to the entryLINK order.

Prior to saving, option selections are ephemeral, allowing the user to consider option changes without committing to them.

Application Validation –

Your application must be reviewed and validated by ProVia before the API Key will be validated for saving.

This application review will include –

- Overall application user interface
- Add Unit, Delete Unit and Change Quantity
- Visual display of
  - Product Description
  - Sizing
  - Image (inside and out)
  - Option selection – including image selectors, Booleans, and decimals
  - Warning listing with icons
  - Warning acknowledgement
- Exit with Cancel

When successfully reviewed, the API key will be unlocked for saving, allowing full functionality.