# Mitchell1 Catalog Driver

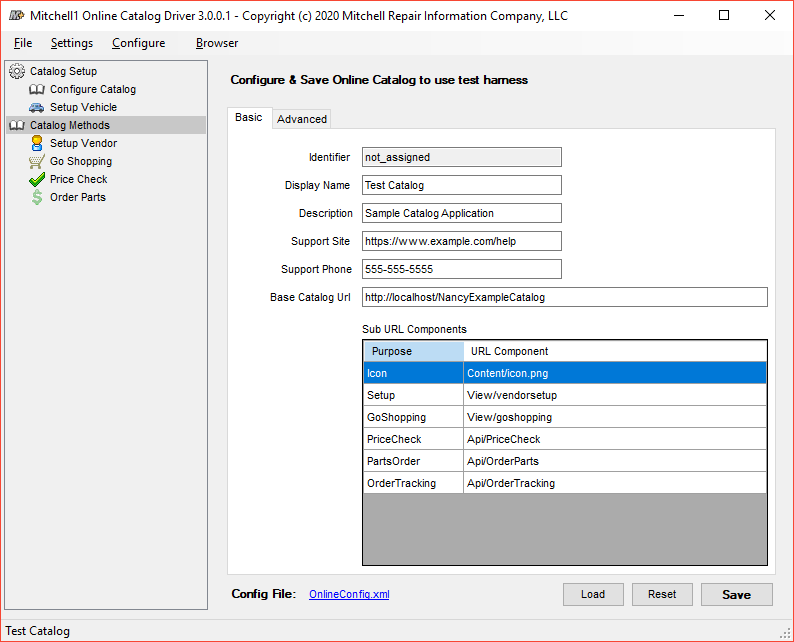
Copyright © 2020 - Mitchell1 Repair Information Company, LLC

## Introduction

This document describes the Mitchell1 Catalog Driver Online edition - which is a stand-alone development tool for testing Online Catalog Integration – for ultimate use with Manager SE. **Important:** The catalog driver is a development aid - all final testing should be performed within the Manager application before release – you will need a Signed XML file to perform that test. The driver is not attempting to replicate all behavior of Manager.

## Basic Usage Guide

Click on “Configure Catalog” in the tree view to configure your catalog. By Default (or if you click “Reset”) it is setup to point at the Example C# Catalog. To start with, you should have also started that sample and leave it running in the background.



Fill in required details (see Online Catalog SDK Doc for more details - ) and click “Save”. Status bar at bottom will update display name of the catalog.

**Advanced** tab contains various settings that control how your catalog behaves. You will want to review those settings and configure applicable settings based on your catalog behavior – should currently work with Manager Product Owner to work on best integration settings for your needs.

**“Load”** button allows you to import an existing configuration (perhaps a signed config you recevied or a file you backed up).

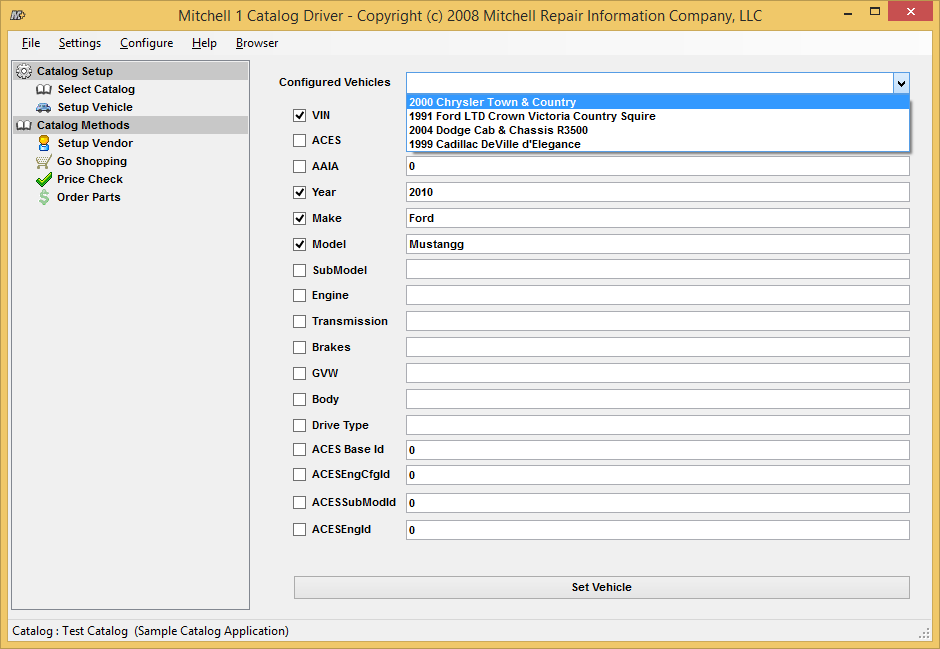
**“Reset”** button restores the settings back to demo mode.

**“Save”** button sames your changes to disk. Once saved, the **Config File** link will light up, and you can click it to quickly access your saved config file (useful for uploading to M1 to be signed).

**Important**: You will need to have a valid URL/site with the above configuration. The SDK includes a very basic C# sample web site/service. Under Sample\Bin\ExampleCatalog.exe (C# Project sources lives under Sample\Source folder). This app will ask for elevated credentials to use http.sys on Windows for a self hosted localhost site (needs elevated credentials as it enables the site for local IPs as well). Once you have your own site up, running this sample is not required, simply point configuration at your URLs.

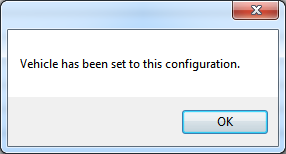
**Option: Setting up a test vehicle to pass along with GoShopping:**

Click on the Setup Vehicle link on the left to configure a vehicle:



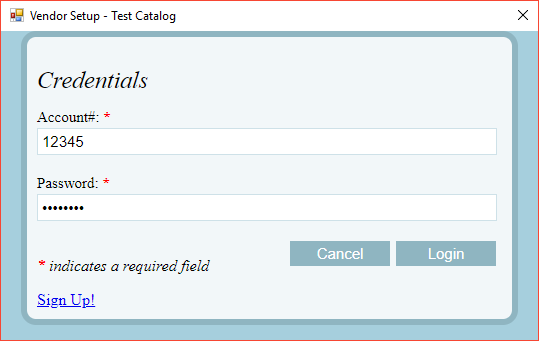
As you can see from the Drop-Down labeled **Configured Vehicles**, there are a few pre-configured vehicles to save you some work.

Click on the **Set Vehicle** button to save your vehicle settings and you should see this dialog:



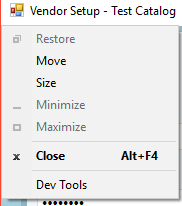
Click on the **Setup Vendor** link on the left to configure a vendor.

Clicking on the Setup Vendor button will bring up the catalog authentication dialog. If you have configured your online catalog correctly, and it is reachable, your vendor setup page will be loaded.



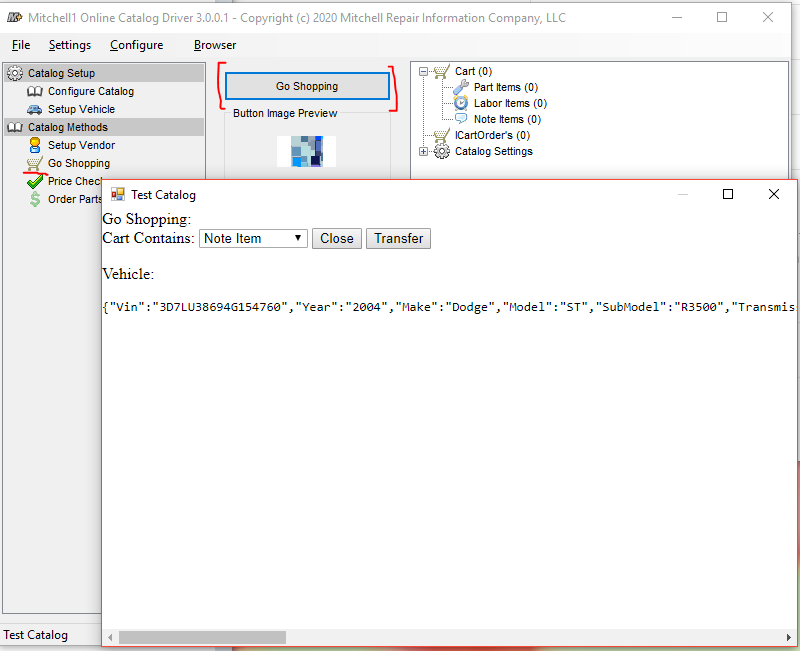
If using the provided Sample Catalog Web Site, Enter the credentials 12345, password and click Login. This will saved the returned object/qualifier locally for later usage in Go Shopping / Price Check / Order Parts.

**Important Notes:**

* The HTML Cancel button will the window from Javascript. Login will close the window, and persist the returned data blob you provide (suggest OAth token, or some other limited scope auth cookie your Server knows how to interpret during later calls)
* Sign Up button illustrates how you can break of this small window to an external browser the client is running (use for help, support, sign up, etc)
* Also, the Window icon (applies to Go Shopping window as well) has an integrated Dev Tools you can use to debug/test
  + 

**Trying out shopping**

In the tree view, select **“Go Shopping”.** Then, click the **“Go Shopping”** button to bring up a catalog window.

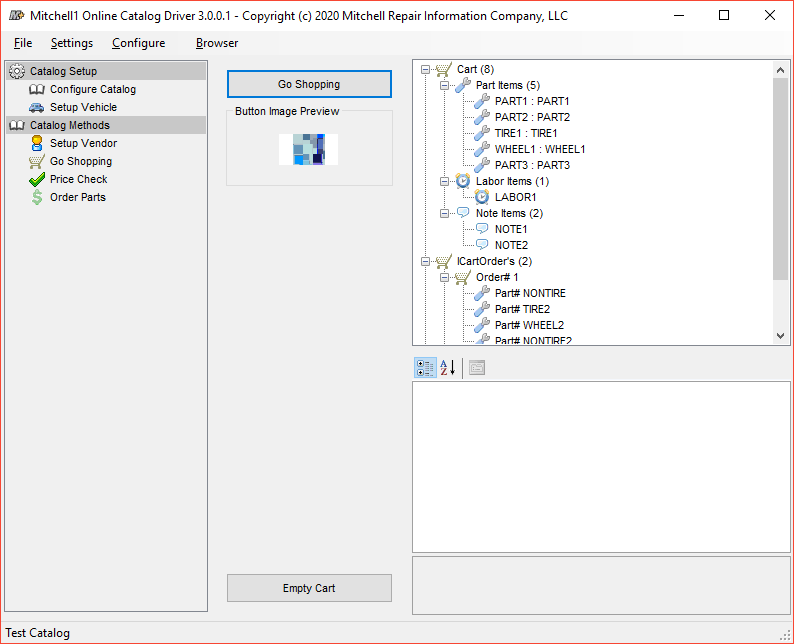


**Important Notes:**

* The image/button display shows the configured Icon/Image for your catalog (from your Icon URL)
* Clicking Close (controlled via your page/javascript) will not transfer anything
* Clicking transfer will transfer what you have populated in your cart. The sample catalog fills the cart with some parts/items and even some directly ordered parts (Important: direct ordering is a good feature to support, but transferring parts without initial order is the most used shop workflow to build up estimates to show to customers – and must be supported as well – using pricecheck/ordering via REST APIs later).
* Refer to the ExpectedHttpErrorCodes.xlsx document on what HTTP errors are expected or not expected from the various integration points

The TreeView in the upper-right allows you to browse the containers that represent the host application (normally Manager SE, but when using the Catalog Driver, it acts as the host). You may click on the part, labor, and note items to view items transferred over from the catalog. You can also click on ICartOrder’s tree view item to expand any parts ordered via GoShopping’s ICartOrder interface. Clicking on the **Empty Cart** button will remove all items from the containers (resetting the cart to its empty state.)

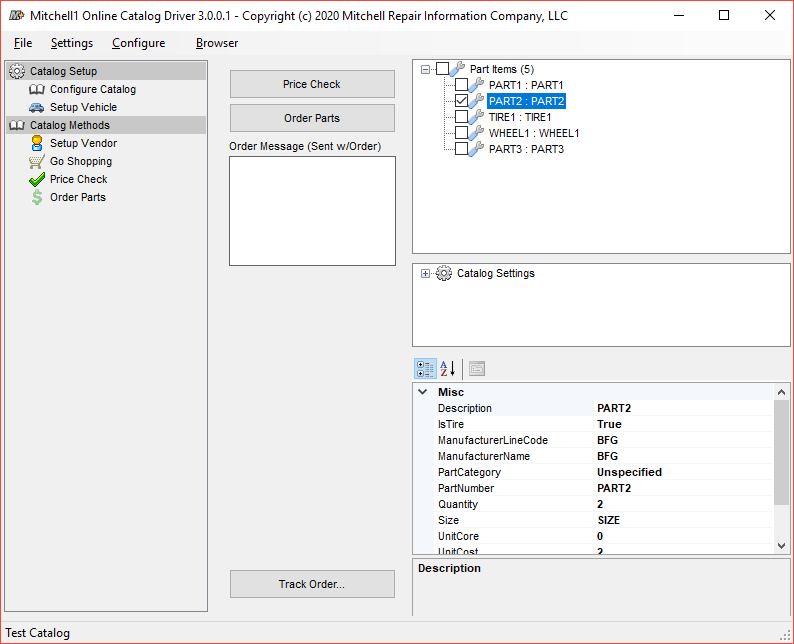
Below is a view of what the Catalog Driver looks like after transferring the part:



The part transferred has been selected and its properties are shown in the bottom right (the property grid).

**Price Checking / Ordering**

At this point you should click on the Price Check link on the left in the tree view:



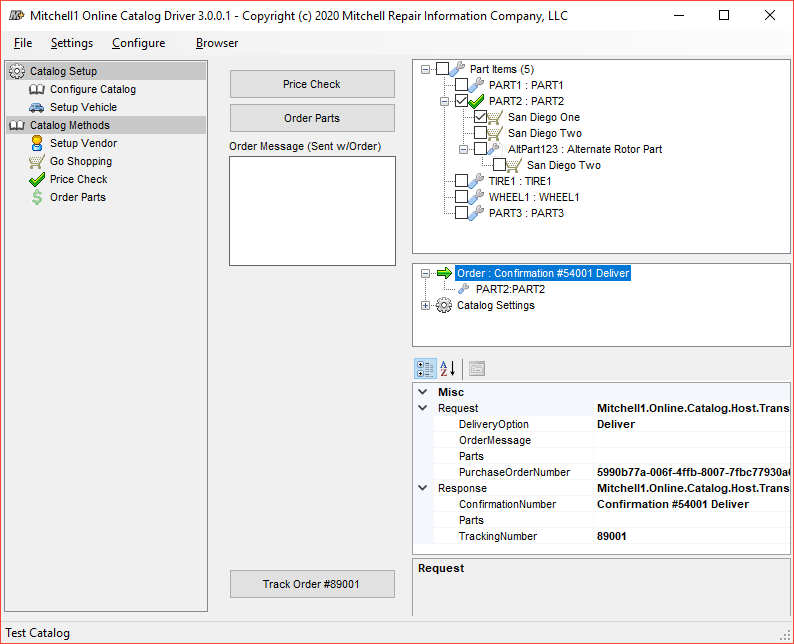
You will need to check the parts you want to Price Check / Order Parts in the TreeView control on the upper-right. This will enable the **Price Check** button. If your catalog properties require that a price check be performed before you can order parts, the **Order Parts** button will still be disabled.

Click on the **Price Check** button to invoke your configured PriceCheck API call to verify the price and availability of parts. Also, alternate locations and their associated quantities may be returned from the web service call. All of this information should be populated in the TreeView control and can be browsed by selecting parts and viewing their properties in the property grid.

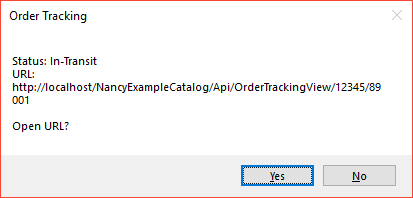
Clicking on the **Order Parts** button will make another call to the configured Order Parts API. This will attempt to order the parts at the locations you selected with the quantity you requested. You should be able to view the status of each part in the order.

## Ordering Tracking

Versions of Manager newer than 8.2.1 support loading a URL from your REST services to bring up your order tracking page. This tracking can be returned via the Go Shopping direct orders, as well as from the Order REST API. If your catalog supports tracking, you would have configured the REST URL for it in the Configure Catalog section, and returned a tracking number. To test that functionality order a part with the tool, the Track Order button should show the tracking # you returned, go ahead and click the track order button.



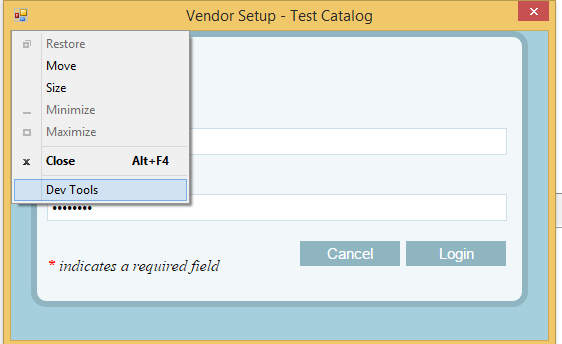
In this demo, a dialog pops up showing both the status string you returned and the URL (clicking Yes will open browser). Manager would present a clickable link to open your tracking page in external browser.



## Command Line Options / Browser Support

The catalog driver has some Mitchell Browser testing tools. In the file menu, under “Browser”, you can load an HTML5 test page, or a simple browser window you can type a URL to navigate to. These allow you to test the internal Web Browser that would be presented for Vendor Setup / Go Shopping web page. You can use this to view the abilities, behaviors, and to debug (with dev tools) your development site as needed.

Additionally, Vendor Setup and Go Shopping Windows have an option in windows’ application menu to launch the tools as well:



Command line flags affect the driver by:

* **--remote-debugging-port=####** (where #### is a valid port number). This specifies a port to startup a remote debugging accessible. At this time, the Mitchell1 Browser is using Chromium/CEF. You can open <http://localhost:####> in Chrome and interact with the browser.
* **Selenium Web testing:** If desired, you can use Selenium via Chrome Driver to run UI automation testing against your site with our browser. Add argument when launching Catalog Driver app - either: **--action=vendor** or **--action=goshopping** (you must have first configured the catalog via normal UI and saved/closed the app). Either argument will directly load a browser with the specified site and will allow Chrome Driver to attach or Selenium tests to run.
* **--browser=[url]** (where [url] is a URL to load). This flag will directly open a browser to said URL – also potentially useful for Selenium web testing.