Get Cimpress Orders

Description
The Get Cimpress Orders App helps fulfiller to receive Enfocus Switch Integration orders present in Cimpress Mass Customization Platform (MCP).

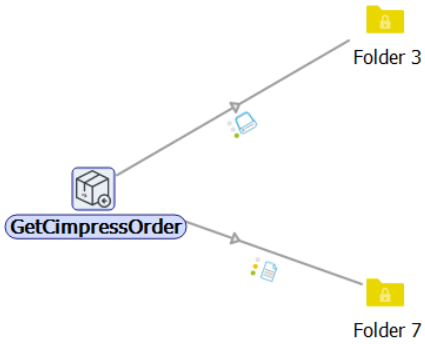
With this app fulfillers can:
- Download details of the order and order-items from MCP.
- Get order details in both xml and json format.
- Automatically download artwork ,preview and jobsheet files related to the order-items in the order.
- Create a new folder to place new orders in a directory that is configured with the app's outgoing data connection.
- Create new folders for every order-item in the respective order folder to hold associated artifacts for the order-item.

Prerequisites
- This app is for fulfillers/suppliers who are registered on the Cimpress Mass Customization Platform (MCP).
- Fulfillers need to set their configuration in Supplier Integration Platform in MCP.

Compatibility
Switch Version Required: Switch 22 and above.
Platform: Windows,Mac

Connections
The Get Cimpress Orders uses the TrafficLight outgoing connection. It has two connections, one to carry data and the other to carry logs. The orders retrieved from MCP are routed to the directory configured with the outgoing connection that carries jobs of type data. The logs generated in the app are routed to the directory linked with the connection that carries logs.

The screen below shows the integration of Get Cimpress Orders into a workflow in Enfocus Switch:



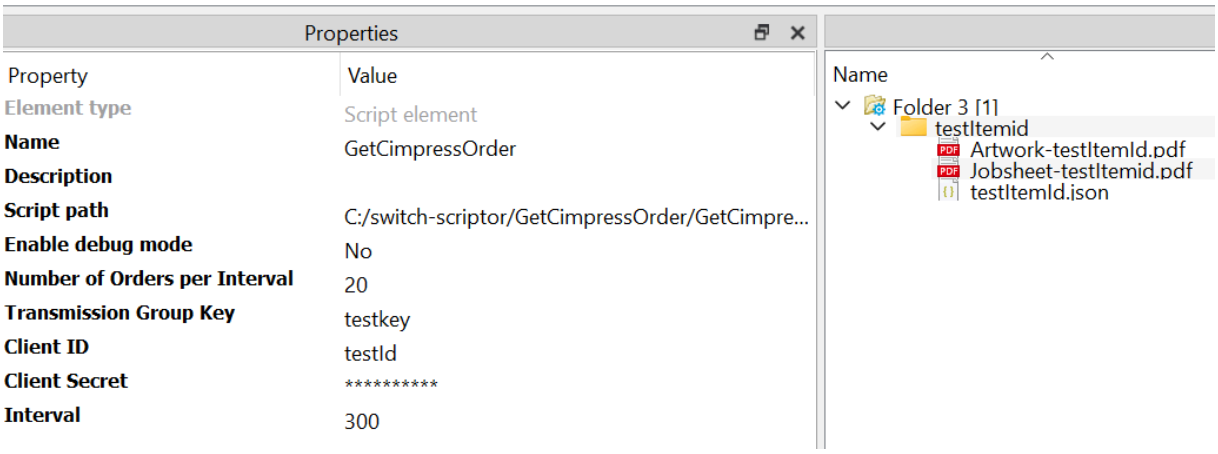


Fig 1: Flow of Get Cimpress Order

Properties detailed info

Properties

Property	Value
Element type	Script element
Name	GetCimpressOrder
Description	
Script path	C:/switch-scriptor/GetCimpressOrder/GetCimpre...
Enable debug mode	No
Number of Orders per Interval	20
Transmission Group Key	testkey
Client ID	testId
Client Secret	**********
Interval	300

Fig 2: Properties Pane

Flow elements properties

- Number of Orders per interval:
  - Number of orders you want to poll in one flow
- Transmission Group Key
  - A unique Identifier for the fulfillers using Enfocus Switch Integration, which will help them to fetch their specific orders from the MCP. This also represents Supplier Integration Configuration, group.
- Client Id
  - The identifier of the client who has access to receive MCP orders for the fulfiller.
- Client Secret
  - The secret is linked to the Client ID
- Interval
  - The time interval in seconds between successive runs of the app. To avoid overlapping two successive app runs, an interval of 10 minutes is recommended.

A sample order detail in JSON Format is given below:


```
{
  "orderMetadata": {
    "orderId": "sampleOrderId",
    "createdDate": "2023-10-31T10:17:32.721+00:00",
    "promisedArrivalDate": "2023-11-29",
    "merchantId": "internal-test-merchant",
    "merchantOrderId": "20231031091007-Ram-Krishna",
    "merchantOrderSupport": {
      "email": "sampleEmail",
      "phoneNumber": "1234567"
```

```
},
"destinationAddress": {
  "country": "IN",
  "postalCode": "12334",
  "stateOrProvince": "State",
  "city": "Bengaluru",
  "company": null,
  "firstName": "Ram Krishna",
  "lastName": "Jha",
  "street1": "",
  "street2": ""
},
"consigneeAddress": {},
"items": [
  {
    "itemId": "testItemId",
    "isTestItem": "false",
    "sku": "Test-sku",
    "orderId": "testOrderId",
    "quantity": "100",
    "name": "Bajirao-Flyer",
    "description": "",
    "document": "documentLink",
    "productManufacturingData": {
    "productCode": "TestCode123",
    "fulfillerProductId": "TestId123",
    "ProductionLine": "1A"
    },
    "merchantInformation": {
      "itemId": "1a0f1873-5138-4d1d-a2d9-c616d7741641",
      "productName": "Bajirao-Flyer"
    },
    "shippingPlans": {
      "expectedCarrierServiceKey": "",
      "expectedCarrierServiceName": "",
      "expectedShipTime": "",
      "expectedShipTimeIsLate": ""
    },
    "deliveryDetails": [
      {
        "type": "Standard",
        "quantity": "100",
        "destinationAddress": {
        "city": "Bengaluru",
        "company": null,
          "firstName": "Ram Krishna",
          "lastName": "Jha",
          "street1": "RMZ Latitude 11th Floor",
          "street2": "Hebbal",
          "country": "IN",
```

```
                "postalCode": "122345",
                "stateOrProvince": "State",
                "isPOBox": null,
                "isResidential": null
              }
            }
          ]
        }
      ],
      "shippingLabelDetails": {
        "returnAddress": {}
      }
    }
  }
```


A sample order detail in XML format is given below:


```
<?xml version="1.0" encoding="UTF-8"?>
<orderMetadata>
  <orderId>TestOrderId</orderId>
  <createdDate>2023-10-31T10:17:32.721+00:00</createdDate>
  <promisedArrivalDate>2023-11-29</promisedArrivalDate>
  <merchantId>internal-test-merchant</merchantId>
  <merchantOrderId>TestMerchantId </merchantOrderId>
  <merchantOrderSupport>
    <email>CTCommerceTribe@cimpress.com</email>
    <phoneNumber>+18675309</phoneNumber>
  </merchantOrderSupport>
  <destinationAddress>
    <country>IN</country>
    <postalCode>123455</postalCode>
    <stateOrProvince>State</stateOrProvince>
    <city>Bengaluru</city>
    <company>null</company>
    <firstName>Ram Krishna</firstName>
    <lastName>Jha</lastName>
    <street1>StreetName</street1>
    <street2>Street Name 2</street2>
  </destinationAddress>
  <consigneeAddress>
  </consigneeAddress>
  <items>
    <item>
      <itemId>testItemId</itemId>
      <isTestItem>false</isTestItem>
      <sku>testSku</sku>
      <orderId>testOrderId</orderId>
      <quantity>100</quantity>
      <name>Bajirao-Flyer</name>
      <description></description>
```

```
    <document>documentLink </document>
    <productManufacturingData>
      <productCode>TestCode123</productCode>
      <fulfillerProductId>TestId123</fulfillerProductId>
      <ProductionLine>1A</ProductionLine>
    </productManufacturingData>
    <merchantInformation>
      <itemId>ItemId </itemId>
      <productName>Bajirao-Flyer</productName>
    </merchantInformation>
    <shippingPlans>
      <expectedCarrierServiceKey></expectedCarrierServiceKey>
      <expectedCarrierServiceName></expectedCarrierServiceName>
      <expectedShipTime></expectedShipTime>
      <expectedShipTimeIsLate></expectedShipTimeIsLate>
    </shippingPlans>
    <deliveryDetails>
      <deliveryDetail>
        <type>Standard</type>
        <quantity>100</quantity>
        <destinationAddress>
          <city>Bengaluru</city>
          <company>null</company>
          <firstName>Ram Krishna</firstName>
          <lastName>Jha</lastName>
          <street1>Street </street1>
          <street2>Street2</street2>
          <country>IN</country>
          <postalCode>12345</postalCode>
          <stateOrProvince>State</stateOrProvince>
          <isPOBox>null</isPOBox>
          <isResidential>null</isResidential>
        </destinationAddress>
      </deliveryDetail>
    </deliveryDetails>
  </item>
 </items>
 <shippingDetails>
  <returnAddress>
  </returnAddress>
 </shippingDetails>
</orderMetadata>
```

Sample Logs Body we get in Get Cimpress Orders App


```
[
  {
    "level": "info",
    "timestamp": "Thu, 02 Nov 2023 06:13:05 GMT",
    "message": "Get Orders App started."
  },
  {
    "level": "info",
    "timestamp": "Thu, 02 Nov 2023 06:13:06 GMT",
    "message": "Fetching downloadables from order."
  }
]
```


Outgoing connections properties

NA

Versions:

Version #	Updates
Version 1	New App in Enfocus Switch Store, which helps fulfillers receive orders from the Cimpress Mass Customization Platform.
Version 2	Added support for Cimpress Clients to be used for authorization
Version 2.1	Enabled file directory creation for the Mac system
Version 2.2	Improved logging
Version 2.3	Enabled item level file structure for multiple items and made fetching number of orders in one flow configurable.