

OIPS Specification – Public API

8th of December, 2004 – Version 2.1.1

This is a specification of the method to send orders directly to Colorplaza System.

Change History:

<i>When</i>	<i>Who</i>	<i>What</i>
20.11.00	Virtanen, Jokitalo	Initial version
02.09.01	Virtanen, Fredembach, Jokitalo	Revised for ArcSoft
07.11.01	Virtanen	Changes for upload
10.10.02	Virtanen	Updated with currently used return codes
20.10.02	Virtanen	Added new calls: AddPromoCode and RefreshSession
01.07.03	Virtanen	Added new upload calls
06.10.03	Jokitalo, Virtanen	New calls: GetServices and SetService
06.11.03	Virtanen, Artimo, Jokitalo	Preparation for version 2.0
12.05.03	Virtanen, Jokitalo	Version 2.0 beta released
13.01.04	Virtanen, Jokitalo	Version 2.0 final
21.01.04	Virtanen	Version 2.1 Minor feature enhancement
08.12.04	Virtanen	Some new status codes and return values added

Table of Contents

Change History.....	2
1. Overview and Introduction.....	5
1.1 Data Format in Data Exchange.....	5
1.1.1 HTTP and HTTPS queries.....	5
1.1.2 Query replies.....	5
1.1.3 Authorization and Content Security.....	5
1.1.4 Protocol.....	5
1.1.5 Amounts (currency fields).....	5
1.1.6 Use of Language with Text and Name fields.....	6
1.1.7 Status codes.....	6
1.2 Order Process	6
1.2.1 Starting the Transaction.....	6
1.2.2 Sessions in Order Process.....	6
1.2.3 Credit Cards.....	7
1.2.4 Adding a Product into the Order.....	7
1.2.5 Retrieving Order Summary.....	7
1.2.6 Promotion codes.....	7
1.2.7 Ending the Transaction.....	7
1.2.8 Typical Order Process.....	7
1.3 Username, password and user handling.....	7
1.4 Version Control.....	8
1.4.1 OIPS Version Control.....	8
1.4.2 Client Software Version Control.....	8
2. Public API.....	9
2.1 GetServer.....	9
2.2 GetTestServer.....	9
2.3 GetCountries.....	10
2.4 GetServices.....	11
2.5 SessionLogin.....	12
2.6 SessionLogout.....	13
2.7 SetLanguage.....	14
2.8 GetCountryName.....	14
2.9 GetProducts.....	15
2.10 GetProductGroups	17
2.11 GetPaymentMethods	18
2.12 GetDeliveryMethods.....	19
2.13 GetAllOptions.....	19
2.14 GetCustomer.....	22
2.15 NewCustomer.....	24
2.16 UpdateCustomer.....	25
2.17 CheckPromoCode.....	27
2.18 CalcOrder	28
2.19 RefreshSession.....	29
2.20 StartOrder.....	30
2.21 AddOrderAddress.....	31
2.22 AddOrderItems.....	32
2.23 ClearOrder.....	34
2.24 VerifyCC.....	34
2.25 SetDeliveryMethod.....	35
2.26 SetPaymentMethod.....	36

2.27 ConfirmOrder	37
2.28 CancelOrder	37
2.29 GetImageServer	38
3. Image Server.....	39
3.1 Upload Image(s).....	39
3.1.1 Parameters.....	39
3.1.2 RFC 1867 Data Stream.....	40
3.1.3 XML Reply.....	40
Appendix A - Order flow example for webupload.....	43
Appendix B - Order flow example for client sw.....	45
Appendix C – OIPS API License.....	47
Appendix D – Reference implementation.....	49

1. Overview and Introduction

This document describes the **Order Interface for Print Service (OIPS)** used by Colorplaza to receive order and image information. Interface specified in this document can be easily used by Album Providers and others to implement print functions into their applications.

Warning: This specification is subject to change.

1.1 Data Format in Data Exchange

1.1.1 HTTP and HTTPS queries

When querying Colorplaza's OIPS server using HTTP or HTTPS protocol all parameter values should be first UTF-8 encoded and then urlencoded (according to RFC 1738) to ensure correct results.

1.1.2 Query replies

Data format to be used is XML 1.0, as a text based standard it is easy to generate. When using XML we could easily solve e.g. issues like character encoding. XML format is used in replies when querying information.

XML 1.0 Specifications can be found at: <http://www.w3.org/XML/>

Reply fields are encoded, as specified by XML, in UTF-8.

1.1.3 Authorization and Content Security

HTTP basic authentication with username and password on URL can be used. SSL connection should be used when sensitive information is sent.

1.1.4 Protocol

During order and payment process HTTP or HTTPS queries are sent and XML responses are received.

Queries and replies are explained in greater detail in later in this document.

1.1.5 User-agent

The HTTP User-Agent header is stored at OIPS SessionLogin where it should be present. The format of the User-Agent string should be following:

```
<OIPS lib>/<version> <character_set> (<win socket dll>; <browser>; <.net>;  
<OS version>; <Screen resolution>; <GUI program>/<version>/<lang_version>)
```

If some fields are not relevant for the client program or operating system in question they can be left empty.

Examples:

ColorPlaza.Net/1.00.95 Unicode (WinINET; IE 6.0.2800.1106; ; Windows XP 5.1.2600 (Service Pack 1) Workstation (Home Edition); 800x600; ColorPlaza.WinUI/1.00.95/FRA)
ColorPlaza.Net/1.00.100 Unicode (WinINET; IE 6.0.2900.2180; .NET Framework v1.1; Windows XP 5.1.2600 (Service Pack 2) Workstation (Home Edition); 1280x1024; ColorPlaza.WinUI/1.00.100/FRA)

1.1.6 Amounts (currency fields)

When currencies are sent, they are sent without decimal notation and last two digits represents decimals (pennies) e.g. <Amount>150</Amount> <Currency> DEM </Currency> equals 1.50 DEM.

1.1.7 Use of Language with Text and Name fields

Text and Name fields are sent according the language that has been set, and are UTF-8 encoded.

1.1.8 Status codes

All OIPS queries return a numerical status code. Status code 0 (zero) means allways that query was successful. Non-zero values indicate that input values were incomplete or invalid, or that a failure occurred during the execution of the query. No error description is given with status code. The status code values are briefly explained in this document. If application wants to provide some error descriptions to the customer, the application developer has to compile them according to the information in the document. As a recommendation, application should use some general error descriptions, that can be used even in case of unknown status code. All the problems and inconsistencies with OIPS API can be reported by e-mail to oips@colorplaza.com.

Many of the status code values have different meaning in different queries, but the basic error situations are consistent. Status code 99 allways means that database connection has failed. Status code 100 allways means that session is expired. Status codes from 1 to 10 are reserved to signal missing input parameters.

1.2 Order Process

To begin the order process, Client needs to call the “GetServer” query to receive the correct server and path where it should send all subsequent queries. This query must be made at the beginning.

Note: It could be that different countries are served by the same server but not necessarily, so at the beginning of the process “GetServer” should be used.

1.2.1 Starting the Transaction

When Client would like to start transaction it should call “SessionLogin” to receive a session id.

1.2.2 Sessions in Order Process

Session id is a 32 character long unique string. Session id should be used in all following calls to OIPS.

Sessions are automatically closed after 30 minutes of inactivity or when “ConfirmOrder” query is received. Otherwise “SessionLogout” can be used to close the session.

1.2.3 Credit Cards

Credit card information is entered into the order with “VerifyCC” query, if credit card is successfully cleared the result code is added into session.

1.2.4 Adding a Product into the Order

Products are added into the order with “AddOrderItems” query. It can be called many times to and add more products to order. “AddOrderItems” will return order summary output. The whole order can be cleared with “ClearOrder” query or starting the order again.

1.2.5 Retrieving Order Summary

Order summary information that is returned after “AddOrderItems” query could be easily produced using “CalcOrder” query with the same parameters. The difference is that then no products are stored into the session or order. When “CalcOrder” is used without parameters it will return the summary information of the products already stored in the order with “AddOrderItems”.

1.2.6 Promotion codes

The promotion codes can be given as parameter to CalcOrder to check the availability of the promotion to the customer owning the session. There is also specialized query “CheckPromoCode”. A description what the promotion is offering is returned, if the promotion code is valid. Because promotions can have conditions per customer, it is necessary that the customer is identified before calling “CheckPromoCode”.

1.2.7 Ending the Transaction

Order is finally registered when “ConfirmOrder” query is made, after which the current session is deleted.

1.2.8 Typical Order Process

Please see Appendix A and B for detailed examples.

1.3 Username, password and user handling

Different platforms and applications are using OIPS. Some application requires user login some just need to have access the customer data. Same user needs to have access to the Print Services via offline client, albums and webupload. Of course it is preferred that we recognize the user and 'account' via her/his own customer id. Unfortunately this not always possible.

Username as concept means: email address, user created nick, or 'login key'. Login key is 32 character long unique string. Email address given by user or nick names created by user should be used in web services like albums and webupload. Login key eg. with offline client.

Example:

Customer registers at Webupload, (s)he selects her/his username (or application use email address as a username) and user selects password.

“NewCustomer” will be called with username and password, OIPS checks that this username is valid and if it exists it will now allow a new login entry. If username already exists application should have the possibility to send username and password to the user eg. via email (using “SendPassword”). If username is unique, OIPS will try to find if this user already has customer record. Match will be made between email, lastname, firstname, addr1, zip, city and country. If exact match has been found, a new login entry is created and existing customer number will be used in this entry in the login table. If more than one exact match has been found, customer record that has the most recent purchases will be used. If no matches are found, new customer will be created.

After this Customer registers with offline client, “NewCustomer” will be called without username and password. OIPS checks if there is exact match for the customer records provided by client software, same logic will be used as above. OIPS will generate 32 character long login key and password that are returned back to client software. These keys should be used by client software in any further calls.

The end result is that there is two entries in the login table that are pointing to the same customer. So customer could have more login key entries than one, but there cannot be more email/nick entries than one.

1. 4 Version Control

1.4.1 OIPS Version Control

OIPS tries to keep backward compatibility but it is not always possible. Therefore it is possible to define what version of OIPS is preferred by the Client in giving version info at the “GetServer” call. Note: version 1.0 of OIPS is not supported.

1.4.2 Client Software Version Control

The software version is checked every time the client starts a new OIPS session. The OIPS will inform if a newer version of the software is available. Client should also support a forced update system in which case the use of the old program is banned. Details of the version control are explained in Section 2.5.1.

2. Public API

2.1 GetServer

Returns URL and Path for the server where all subsequent queries should be made.

`http://oips.colorplaza.com/GetServer.cpl?ecomid=nnn...`

Parameters:

Name	Description
version	OIPS version to be used (optional). If not given default will be selected. 20 = version 2.0 [default]
ecomid	ecommerce id (optional), for future use

Return values:

Name	Description
HTTPServer	The address of the OIPS server
HTTPath	The path name on the server that has to be used
Status	0=ok

Example:

```
<?xml version="1.0" encoding="utf-8" ?>
<GetServer>
  <Status>0</Status>
  <HTTPServer>orders.colorplaza.com</HTTPServer>
  <HTTPath>/oips/</HTTPath>
</GetServer>
```

2.2 GetTestServer

Same as GetServer, but this call return the test server address that can be used for testing. Returns URL and Path for the server where all subsequent queries should be made.

`http://oips.colorplaza.com/GetTestServer.cpl?ecomid=nnn...`

Parameters:

Name	Description
version	OIPS version to be used (optional). If not given default will be selected. 20 = version 2.0 [default]
ecomid	ecommerce id (optional), for future use

Return values:

Name	Description
HTTPServer	The address of the OIPS server
HTTPPath	The path name on the server that has to be used
Status	0=ok

Example:

```
<?xml version="1.0" encoding="utf-8" ?>
<GetTestServer>
  <Status>0</Status>
  <HTTPServer>213.193.107.156</HTTPServer>
  <HTTPPath>/oips-test20</HTTPPath>
</GetTestServer>
```

2.3 GetCountries

Return a list of all available countries, their country code and name. This call does not require valid OIPS session.

`http://<HTTPServer><HTTPPath>GetCountries.cpl`

Parameters:

Name	Description
oemid	OEMID provided by Colorplaza. Optional, if used will limit the list of countries to those where the service is available.

Return values:

Name	Description
Code	3 character ISO-3166 country code
CountryName	Country name, international
CountryNameLocal	Country name in local language(s). If there is more than one language the format of this string is eg. de=schweiz, fr=suisse, it=svizzera
Translation	Country name in given language specified in language attribute. If default attribute is Y translation is the default name. Note that CountryNameLocal and Translation are representations of the same data in different form.
Status	0= ok 30= db error, no countries found 99= db connection failed

Example:

```
<?xml version="1.0" encoding="utf-8" ?>
<GetCountries>
  <Status>0</Status>
  <Country>
    <Code>AUT</Code>
    <CountryName>AUSTRIA</CountryName>
    <CountryNameLocal>EN=AUSTRIA, DE=Österreich</CountryNameLocal>
    <Translation language="EN" default="Y">AUSTRIA</Translation>
    <Translation language="DE">Österreich</Translation>
    <Translation language="EN">AUSTRIA</Translation>
  </Country>
  ...
</GetCountries>
```

2.4 GetServices

Return a list of available services, their target country and default language. This call does not require valid OIPS session.

`http://<HTTPServer><HTTPPath>GetServices.cpl?oemid=123&cntry=deu`

Usage Notes:

- 1) if no parameters are given, list of all services from all oemids is returned
- 2) if oemid is given, list of services for specified oemid is returned
- 3) if oemid and country are given, service available for specified oemid in country in question is returned

Parameters:

Name	Description
oemid	OEM id provided by Colorplaza (Optional)
cntry	3 character ISO-3166 country code. (Optional)

Return values:

Name	Description
Seid	Service id
Name	Name of the service
Country	3 character ISO-3166 country code. Target country of the service. Note: If country given in the parameters is other country than "localized country" INT or INX
DftLangId	2 character ISO-639 language code. Default language of the service.
LangId	Possible other language. 2 character ISO-639 language code.
ServiceDescriptionText	Longer description of the service, formatted as HTML
ServiceIconURL	Fully qualified URL for service icon (optional)
ServiceLogoURL	Fully qualified URL for service logo (optional)
ServiceFAQURL	Fully qualified URL for service faq page (optional)
ServiceContactURL	Fully qualified URL for service contact page (optional)
ServiceHomePageURL	Fully qualified URL for service home page (optional)
Status	0=ok 1= required parameter oemid is missing 5= invalid oemid 10= configuration error: no services found 30= db error reading services

Example:

```
<?xml version="1.0" encoding="utf-8" ?>
<GetServices>
  <Status>0</Status>
  <Service>
    <Seid>1234</Seid>
    <Name>Colormailer CH</Name>
    <Country>CHE</Country>
    <Languages>
      <DftLangId>DE</DftLangId>
      <LangId>IT</LangId>
      <LangId>FR</LangId>
    </Languages>
    <ServiceDescriptionText>Some blabla here</ServiceDescriptionText>
    <ServiceIconURL>http://colorplaza.net/client/icon.gif
    </ServiceIconURL>
    <ServiceLogoURL>http://colorplaza.net/client/logo.jpg
    </ServiceLogoURL>
    <ServiceFAQURL>http://www.colormailer.ch/faq.cpl?lang=de
    </ServiceFAQURL>
    <ServiceContactURL>http://www.colormailer.ch/contact.cpl?lang=de
    </ServiceContactURL>
    <ServiceHomePageURL>http://www.colormailer.ch/home.cpl?lang=de
    </ServiceHomePageURL>
  </Service>
  ...
</GetServices>
```

2.5 SessionLogin

Starts a session and returns a session id that has to be used in all subsequent queries. Set service id (seid) for session based on the return of GetServices. This query has to be

done thru HTTPS.

`https://<HTTPServer><HTTPPath>SessionLogin.cpl?login=test&pw=xx&seid=...`

Note1: Session is automatically closed: after 30 minutes inactivity, or when ConfirmOrder query is successfully made.

2.5.1 Client version control

The software version is checked via a two-way upgrade mechanism each time a customer logs in : If a newer version of the software is found, an upgrade is suggested. In that case, the ClientUpgradeURL parameter contains the URL that links to a HTML page, where the program can be downloaded. It is up to the customer to download and install the newer version, but it is not required in order to use the service.

Old and/or bad versions of the software can be blocked by Colorplaza. In that case the ForceUpdate parameter is set and the client program will show a warning message and a link to the update package. When update is forced, the ClientUpgradeURL will contain a URL to update information page, NewClientSetupURL will contain a link to the update install package and NewClientPackageSizeBytes will show the update package size. After showing the update message the client should exit and not let the customer use the service before performing the software update.

Parameters:

Name	Description
login	valid id to log in to system (provided by copla)
pw	valid password (provided by copla)
seid	Service id (from GetServices)
swid	Software id (defined by colorplaza)
swversion	Defined by colorplaza, format like: 1.2.0
langid	language used in this session
oemid	OEM identifier
ipaddr	(optional) used by web applications
useragent	(optional) used by web applications

Return values:

Name	Description
Id	32 char long session identifier
Theme	WebUpload theme parameter
Country	Colorplaza internal service country code
WhatsNewURL	Fully qualified URL for new features page

Name	Description
ClientUpgradeURL	Fully qualified URL to upgrade info page
NewClientSetupURL	Fully qualified URL for client setup program
NewClientPackageSizeBytes	Size of the client package in bytes
ForceUpdate	Update is obligatory, value Y or N
ImageServer	Image server host name or address
Status	0= ok 1= login missing 2= pw missing 3= seid missing 4= swid missing 5= swversion missing 6= langid missing 7= oemid missing 9= undefined swid 11= login rejected 12=no service found for cntry/lang 15= configuration error, imageserver not found 20= configuration error, no default delivery defined 21= db error creating new session

Example:

```
<?xml version="1.0" encoding="utf-8" ?>
  <SessionLogin>
    <Status>0</Status>
    <Id>32ad89ed34c1d48310ee61ad89ed34c1</Id>
    <WhatsNewURL>http://colorplaza.net/client/ReleaseNotes.html
    </WhatsNewURL>
    <ClientUpgradeURL></ClientUpgradeURL>
    <NewClientSetupURL></NewClientSetupURL>
    <NewClientPackageSizeBytes>1234560</NewClientPackageSizeBytes>
    <ForceUpdate>N</ForceUpdate>
    <ImageServer>imagescluster.colorplaza.com</ImageServer>
  </SessionLogin>
```

2.6 SessionLogout

Closing a session.

`http://<HTTPServer><HTTPPath>SessionLogout.cpl?sid=32ad89e`

Note: In normal operation you do not need this function. The use is optional and even unnecessary when "ConfirmOrder" is used.

Parameters:

Name	Description
sid	session id

Return values:

Name	Description
Status	0= ok 1= required input parameter sid is missing 30= db error in logout 100= session not found

Example:

```
<?xml version="1.0" encoding="utf-8" ?>
<SessionLogout>
  <Status>0</Status>
</SessionLogout>
```

2.7 SetLanguage

Set language for this session. This language is then used with all “Name” and other descriptive output fields.

`http://<HTTPServer><HTTPPath>SetLanguage.cpl?sid=32ad89e...&lang=fr`

Parameters:

Name	Description
sid	session id
langid	2 char ISO-639 language code

Return values:

Name	Description
Status	0= ok 1= required input parameter sid is missing 2= langid is missing 12= no service in specified language 30= db error 100= session not found

Example:

```
<?xml version="1.0" encoding="utf-8" ?>
<SetLanguage>
  <Status>0</Status>
</SetLanguage>
```

2.8 GetCountryName

Query for country name.

`http://<HTTPServer><HTTPPath>GetCountryName.cpl?sid=32ad89e&cntry=nor`

Parameters:

Name	Description
sid	session id
cntry	3 character ISO-3166 country code

Return values:

Name	Description
CountryName	Country name, intrnational name.
CountryNameLocal	Country name in local language. Note this is only available for the main countries served. So if not existing use international.
Status	0=ok 1= required parameter sid is missing 2= required parameter cntry is missing 15= Invalid country code 100=session not found

Example:

```
<?xml version="1.0" encoding="utf-8" ?>
<GetCountryName>
  <Status>0</Status>
  <CountryName>Sweden</CountryName>
  <CountryNameLocal>Sverige</CountryNameLocal>
</GetCountryName>
```

2.9 GetProducts

Returns all available products for the specified service. Optionally a product group parameter can be given so that only the products in that group will be listed.

`http://<HTTPServer><HTTPPath>GetProducts.cpl?sid=32ad89e`

An additional discount price is given if quantity exceeds a specified limit. A discount level is specified by a discount price - quantity limit pair. For example: Price = 23 and MinQty = 100. That means if the total product quantity is greater that or equal to 100, the product unit price will be 23. If there are more that one level, the highest discount level that applies should be selected.

Parameters:

Name	Description
sid	session id
group	Product group id (optional)

Return values:

Name	Description
Pid	Product Id

Name	Description
Name	Name of the product
Price	Product unit price, in minor units
Currency	Currency code
MinXRes	Minimum recommended x resolution
MinYRes	Minimum recommended y resolution
Width	Product width in millimeters
Height	Product height in millimeters (the longer dimension)
WidthHiRes	Product width in 1/100 millimeters
HeightHiRes	Product height in 1/100 millimeters
OptimalDpi	Recommended dpi (dots per inch) resolution for printing
VariableLength	Product has flexible size, 0=no, 1=yes
MinAspectRatio	Minimum aspect ratio, used if VariableLength is enabled
MaxAspectRatio	Maximum aspect ratio, used if VariableLength is enabled
DpiLim1	Dpi Limit of poor quality warning
DpiLim2	Dpi Limit of adequate quality
DpiLim3	Dpi Limit of good quality
Ranking	Product ranking value, list in ascending order
GroupId	Product group Id
ProductGroup	Product group Id, same as GroupId
Default	Indicates if this is default product of the group, Y/N
Status	0=ok 1= required parameter sid is missing 30= No products found. OIPS configuration error 31= No currency found. OIPS configuration error 100=session not found

For every quantity discount level:

Name	Description
Price	Product unit price for this discount level
MinQty	Minimum quantity for the discount

Example:

```
<?xml version="1.0" encoding="utf-8" ?>
<GetProducts>
  <Status>0</Status>
  <Currency>EUR</Currency>
  <Product>
    <Pid>1</Pid>
    <Name>10 x 15cm</Name>
    <Price>35</Price>
    <MinXRes>480</MinXRes>
    <MinYRes>640</MinYRes>
    <Width>101</Width>
    <Height>152</Height>
    <OptimalDpi>300</OptimalDpi>
    <ProductGroup>1</ProductGroup>
  </Product>
</GetProducts>
```

```

    <Ranking>50</Ranking>
    <GroupId>1</GroupId>
    <dpiLim1>80</dpiLim1>
    <dpiLim2>150</dpiLim2>
    <dpiLim3>250</dpiLim3>
    <Default>N</Default>
    <VariableLength>0</VariableLength>
    <MinAspectRatio>1.0000</MinAspectRatio>
    <MaxAspectRatio>1.0000</MaxAspectRatio>
    <WidthHiRes>11400</WidthHiRes>
    <HeightHiRes>15200</HeightHiRes>
    <QuantityDiscount>
      <Level>
        <Price>30</Price>
        <MinQty>50</MinQty>
      </Level>
    </QuantityDiscount>
  </Product>
</Product>
  <Pid>4</Pid>
  <Name>11 x 15cm</Name>
  <Price>38</Price>
  <MinXRes>480</MinXRes>
  <MinYRes>640</MinYRes>
  <Width>114</Width>
  <Height>152</Height>
  <OptimalDpi>300</OptimalDpi>
  <Ranking>10</Ranking>
  <GroupId>1</GroupId>
  <dpiLim1>80</dpiLim1>
  <dpiLim2>150</dpiLim2>
  <dpiLim3>250</dpiLim3>
  <Default>Y</Default>
  <VariableLength>1</VariableLength>
  <MinAspectRatio>1.0000</MinAspectRatio>
  <MaxAspectRatio>1.7710</MaxAspectRatio>
  <WidthHiRes>11400</WidthHiRes>
  <HeightHiRes>15200</HeightHiRes>
  <QuantityDiscount>
    <Level>
      <Price>34</Price>
      <MinQty>100</MinQty>
    </Level>
    <Level>
      <Price>30</Price>
      <MinQty>500</MinQty>
    </Level>
  </QuantityDiscount>
</Product>
</GetProducts>

```

2.10 GetProductGroups

Returns a list of product groups.

<http://<HTTPServer><HTTPPath>GetProductGroups.cpl?sid=32ad>

Parameters:

Name	Description
sid	session id

Return values:

Name	Description
GroupId	Product group id
Name	The name of the product group
Description	Product group description
DefaultProductId	Product id of the default product
Default	Group is default group, Y/N
Ranking	Product group ranking value
Status	0= ok 1= required input parameter sid is missing 30= configuration error, no payment methods defined 100= session expired

Example:

```
<?xml version="1.0" encoding="utf-8" ?>
<GetProductGroups>
  <Group>
    <GroupId>1</GroupId>
    <Name>Photos and Enlargements</Name>
    <Description> ... </ Description >
    <DefaultProductId>4</DefaultProductId>
    <Default>Y</Default>
    <Ranking>1</Ranking>
  </Group>
  <Group>
    <GroupId>2</GroupId>
    <Name>Posters</Name>
    <Description> ... </ Description >
    <DefaultProductId>42</DefaultProductId>
    <Default>N</Default>
    <Ranking>2</Ranking>
  </Group>
</GetProductGroups>
```

2.11 GetPaymentMethods

Returns all available payment methods for the specified country.

`http://<HTTPServer><HTTPPath>GetPaymentMethods.cpl?sid=32ad`

Parameters:

Name	Description
sid	session id

Return values:

Name	Description
Payid	Payment method id
Payname	name of the payment method
PaymentType	0=invoice, 1=cc
Status	0= ok 1= required input parameter sid is missing 30= configuration error, no payment methods defined 100= session expired

Example:

```
<?xml version="1.0" encoding="utf-8" ?>
<GetPaymentMethods>
  <Payment>
    <Payid>1</Payid>
    <Payname>Invoice</Payname>
    <PaymentType>0</PaymentType>
  </Payment>
  <Payment>
    <Payid>4</Payid>
    <Payname>Visa</Payname>
    <PaymentType>1</PaymentType>
  </Payment>
  <Payment>
    <Payid>5</Payid>
    <Payname>MasterCard</Payname>
    <PaymentType>1</PaymentType>
  </Payment>
</GetPaymentMethods>
```

2.12 GetDeliveryMethods

Returns all available delivery methods for the specified country.

<http://<HTTPServer><HTTPPath>GetDeliveryMethods.cpl?sid=32ad>

Parameters:

Name	Description
sid	session id

Return values:

Name	Description
Did	Delivery method id
Name	Name of the delivery method
Default	Y/N indicates if this is the default delivery method
Price	Delivery price in minor units
Currency	Currency code used for delivery price

Name	Description
Status	0=ok 1= required input parameter sid is missing 30= no delivery methods found 32= db error reading currency 100=session not found

Example:

```
<?xml version="1.0" encoding="utf-8" ?>
<GetDeliveryMethods>
  <Status>0</Status>
  <Currency>USD</Currency>
  <Delivery>
    <Did>1</Did>
    <Name>Priority Mail</Name>
    <Price>150</Price>
    <Default>Y</Default>
  </Delivery>
  <Delivery>
    <Did>2</Did>
    <Name>International Express</Name>
    <Price>1200</Price>
    <Default>N</Default>
  </Delivery>
</GetDeliveryMethods>
```

2.13 GetAllOptions

Returns all available products, payment methods and delivery methods for the selected service. The output is almost like GetProducts, GetPaymentMethods and GetDeliveryMethods combined. Additionally a product group parameter can be given so that only the products in that group will be listed.

<http://<HTTPServer><HTTPPath>GetAllOptions.cpl?sid=32ad89e>

Parameters:

Name	Description
sid	session id
group	Product group id (optional)

Return values:

Name	Description
Currency	Currency code used for all prices
ColorCorrection	Automatic color correction is available. 0=no, 1=yes. Colorcorrection parameter could be set per Image using "AddOrderItems".
Theme	WebUpload theme name

Name	Description
Status	0=ok 1= required input parameter sid is missing 30= db error reading currency 31= db error reading products 32= db error reading payment methods 33= db error reading delivery methods 100=session not found

For every product:

Name	Description
Pid	Product id
Pname	Product name
Price	Product unit price
MinXRes	Minimum recommended x resolution
MinYRes	Minimum recommended y resolution
Width	Product width in millimeters
Height	Product height in millimeters
WidthHiRes	Product width in 1/100 millimeters
HeightHiRes	Product height in 1/100 millimeters
OptimalDpi	Optimal dpi for printing
DpiLim1	Dpi Limit of poor quality warning
DpiLim2	Dpi Limit of adequate quality
DpiLim3	Dpi Limit of good quality
Ranking	Product ranking value, list in ascending order
GroupId	Product group id
ProductGroup	Product group id, same as GroupId
VariableLength	Product has variable size; 0=no, 1=yes
MinAspecRatio	Minimum aspect ratio of print, with four decimals
MaxAspecRatio	Maximum aspect ratio of print, with four decimals
Default	Product is the default product, Y/N

For every quantity discount level:

Name	Description
Price	Discounted unit price
MinQty	Minimum quantity for the discount

For every product group:

Name	Description
GroupId	Product group id
Name	Product group name

Name	Description
Description	Product group description
DefaultProductId	Product id of the group's default product
Default	Product Group is default group, Y/N
Ranking	Product Group ranking value

For every payment method:

Name	Description
Payid	Payment method id
Payname	Payment method name
PaymentType	0=invoice, 1=cc

For every delivery method:

Name	Description
Did	Delivery id
Dname	Delivery method name
Dprice	Delivery price
Default	Y or N, indicates if this is the default delivery method

Example:

```
<?xml version="1.0" encoding="utf-8" ?>
<GetAllOptions>
  <Status>0</Status>
  <Currency>CHF</Currency>
  <Theme>cm</Theme>
  <ColorCorrection>1</ColorCorrection>
  <ProductGroups>
    <Group>
      <GroupId>1</GroupId>
      <Name>Photos</Name>
      <Description>...</Description>
      <DefaultProductId>4</DefaultProductId>
      <Default>Y</Default>
      <Ranking>20</Ranking>
    </Group>
    <Group>
      <GroupId>2</GroupId>
      <Name>Posters</Name>
      <Description>...</Description>
      <DefaultProductId>24</DefaultProductId>
      <Default>N</Default>
      <Ranking>40</Ranking>
    </Group>
  </ProductGroups>
  <Products>
    <Product>
      <Pid>110</Pid>
      <Pname>11x15 cm</Pname>
      <Price>35</Price>
      <MinXRes>480</MinXRes>
      <MinYRes>640</MinYRes>
      <Width>114</Width>
      <Height>152</Height>
    </Product>
  </Products>
</GetAllOptions>
```

```

        <OptimalDpi>300</OptimalDpi>
        <GroupId>1</GroupId>
        <ProductGroup>1</ProductGroup>
        <dpiLim1>80</dpiLim1>
        <dpiLim2>150</dpiLim2>
        <dpiLim3>250</dpiLim3>
        <Default>Y</Default>
        <VariableLength>0</VariableLength>
        <MinAspectRatio>1.0000</MinAspectRatio>
        <MaxAspectRatio>1.0000</MaxAspectRatio>
        <Ranking>1</Ranking>
        <WidthHiRes>11400</WidthHiRes>
        <HeightHiRes>15200</HeightHiRes>
        <QuantityDiscount>
            <Level>
                <Price>30</Price>
                <MinQty>200</MinQty>
            </Level>
        </QuantityDiscount>
    </Product>
    <Product>
        <Pid>167</Pid>
        <Pname>13x19 cm</Pname>
        <Price>110</Price>
        <MinXRes>480</MinXRes>
        <MinYRes>640</MinYRes>
        <Width>127</Width>
        <Height>190</Height>
        <OptimalDpi>280</OptimalDpi>
        <GroupId>1</GroupId>
        <ProductGroup>1</ProductGroup>
        <dpiLim1>90</dpiLim1>
        <dpiLim2>160</dpiLim2>
        <dpiLim3>260</dpiLim3>
        <Default>N</Default>
        <VariableLength>1</VariableLength>
        <MinAspectRatio>1.0000</MinAspectRatio>
        <MaxAspectRatio>1.7710</MaxAspectRatio>
        <Ranking>12</Ranking>
        <WidthHiRes>12700</WidthHiRes>
        <HeightHiRes>19000</HeightHiRes>
        <QuantityDiscount>
            <Level>
                <Price>100</Price>
                <MinQty>100</MinQty>
            </Level>
            <Level>
                <Price>90</Price>
                <MinQty>500</MinQty>
            </Level>
        </QuantityDiscount>
    </Product>
</Products>
<PaymentMethods>
    <Payment>
        <Payid>4</Payid>
        <Payname>Invoice</Payname>
        <PaymentType>0</PaymentType>
    </Payment>
    <Payment>
        <Payid>1</Payid>
        <Payname>Visa</Payname>
        <PaymentType>1</PaymentType>
    </Payment>
</PaymentMethods>
<DeliveryMethods>
    <Delivery>

```



```

        <Did>1</Did>
        <Dname>Priority Mail</Dname>
        <Dprice>395</Dprice>
        <Default>Y</Default>
    </Delivery>
    <Delivery>
        <Did>2</Did>
        <Dname>DHL express</Dname>
        <Dprice>1690</Dprice>
        <Default>N</Default>
    </Delivery>
</DeliveryMethods>
</GetAllOptions>

```

2.14 GetCustomer

Retrieve customer information with customer username and password.
All customer queries are made with https.

`https://<HTTPServer><HTTPPath>GetCustomer.cpl?sid=32ad89e&username=bob@domain.dom&userpw=xyz`

Parameters:

Name	Description
sid	session id
username	Username
userpw	User password

Return values:

Name	Description
Title	Title (optional)
First	First name
Last	Last name
Addr1	Address line 1
Addr2	Address line 2
City	City
Zip	Postal code
Country	Three letter country code
CountryName	Country name (in local language, if available)
State	State or Province
Email	Email address
Remember	Remember me option, Y=yes or N=no
Phone	Telephone number (optional)
Fax	Fax number (optional)
NoEmail	No additional emails from Colorplaza, value: Y / N (optional)

Name	Description
Gender	Gender; 0=Female, 1=Male
Username	Username
Userpw	User password
Status	0=ok 1= sid is missing 2= username is missing 3= passwd is missing 5= invalid login 20= error reading customer data 30= session update failed 100=session not found

Example:

```
<?xml version="1.0" encoding="utf-8" ?>
<GetCustomer>
  <Status>0</Status>
  <Title>Mr.</title>
  <First>Test</First>
  <Last>User</Last>
  <Addr1>Rue Centrale 1</Addr1>
  <Addr2></Addr2>
  <City>Lausanne</City>
  <Zip>1002</Zip>
  <CountryCode>CHE</CountryCode>
  <CountryName>SUISSE</CountryName>
  <State>VD</State>
  <Email>user@domain.com</Email>
  <Remember>Y</Remember>
  <NoEmail>N</NoEmail>
  <Phone>+41-21-9233506</Phone>
  <Fax>+41-21-9233513</Fax>
  <Gender>1</Gender>
  <Username>test</Username>
  <Userpw>secret</Userpw>
</GetCustomer >
```

2.15 NewCustomer

Create a new customer to OIPS.

<https://<HTTPServer><HTTPPath>NewCustomer.cpl?sid=32ad &first=...>

Note1: If username and userpassword are not given a new pair of username and userpassword will be generated. Basically online applications should use this call with username and password and offline clients without. For more information please read section 1.3.

Note2: Both username and userpassword has to be given or not given at all.

Parameters:

Name	Description
sid	Session id
username	Username (optional) pls. see Note1&2 Max. 200 characters long
userpw	User password (optional) pls. see Note1&2 Max. 32 characters
title	Title (optional) Max. 20 characters
first	First name Max. 128 characters
last	Last name Max. 128 characters
addr1	Address line 1 Max. 128 characters
addr2	Address line 2 (optional) Max. 128 characters
city	City Max. 128 characters
zip	Postal code (optional if country is IRL) Max. 35 characters
cntry	Three letter country code
state	State or province (optional) Max. 50 characters
email	E-mail address Max. 128 characters
remember	Remember me option, Y / N
noemail	No additional emails from Colorplaza, value: Y / N Parameter is optional
phone	Telephone number, Max. 50 characters (optional)
fax	Fax number, Max. 50 characters (optional)
gender	Gender; 0=female, 1=male

Return values:

Name	Description
Username	Username entered
Userpw	Password entered

Name	Description
Status	0=ok 1= sid is missing 2= first name is missing or is empty 3= last name is missing or is empty 4= addr1 field is missing or is empty 5= city is missing or is empty 6= country code is missing or is empty 7= zip is missing 8= remember is missing 9= email is missing 10= username given, but userpw is missing 11= username already used 12= parameter first exceeds size limit 13= parameter last exceeds size limit 14= parameter addr1 exceeds size limit 15= parameter addr2 exceeds size limit 16= parameter city exceeds size limit 17= parameter state exceeds size limit 18= parameter zip exceeds size limit 19= parameter email exceeds size limit 20= parameter title exceeds size limit 21= parameter username exceeds size limit 22= parameter userpw exceeds size limit 24= customer does not belong to selected service 25= parameter phone exceeds size limit 26= parameter fax exceeds size limit 29= parameter reading error 30= db error updating session 33= db error creating customer 41= customer is already identified in this OIPS session, use UpdateCustomer instead 100=session not found

Example:

```
<?xml version="1.0" encoding="utf-8" ?>
<NewCustomer>
  <Status>0</Status>
  <Username>testuser</Username>
  <Userpw>secret</Userpw>
</NewCustomer>
```

2.16 UpdateCustomer

Update existing customer data in OIPS. There are same field length limitations that in NewCustomer query, see 2.14.

`https://<HTTPServer><HTTPPath>UpdateCustomer.cpl?sid=32ad &first=...`

Note1: you cannot change username and/or password with this query. Username and password are used as a key to find the customer.

Parameters:

Name	Description
sid	Session id
username	Username, see Note1
userpw	User password, see Note1
title	Title (optional)
first	First name
last	Last name
addr1	Address line 1
addr2	Address line 2 (optional)
city	City
zip	Postal code
cntry	Three letter country code
state	State or province (optional)
email	E-mail address (optional)
remember	Remember me option, Y or N
noemail	No additional emails from Colorplaza, value: Y / N Parameter is optional
phone	Telephone number (optional)
fax	Fax number (optional)
gender	Gender; 0=female, 1=male

Return values:

Name	Description
Username	Username
Userpw	Password

Name	Description
Status	0=ok 1= sid is missing 2= first name is missing or is empty 3= last name is missing or is empty 4= addr1 field is missing or is empty 5= city is missing or is empty 6= country code is missing or is empty 7= zip is missing 8= username is missing 9= userpw is missing 10= invalid username-userpw combination 11= username is different than owner of this session 12= parameter first exceeds size limit 13= parameter last exceeds size limit 14= parameter addr1 exceeds size limit 15= parameter addr2 exceeds size limit 16= parameter city exceeds size limit 17= parameter state exceeds size limit 18= parameter zip exceeds size limit 19= parameter email exceeds size limit 20= parameter title exceeds size limit 21= parameter phone exceeds size limit 22= parameter fax exceeds size limit 32= database error updating customer 33= database error updating session 100= session expired

Example:

```
<?xml version="1.0" encoding="utf-8" ?>
<UpdateCustomer>
  <Status>0</Status>
  <Username>testuser</Username>
  <Userpw>secret</Userpw>
</UpdateCustomer>
```

2.17 CheckPromoCode

Check promo code validity. Returns a small description about the promotion.

`http://<HTTPServer><HTTPPath>CheckPromoCode.cpl?sid=4324abc&promo=FREE21`

This query will not store the promocode and it has to be given in "StartOrder". This query just checks the existence of the promocode.

Parameters:

Name	Description
sid	session id
username	Username
promo	Promotion code

Return values:

Name	Description
PromoDescription	Promotion description
Status	0=ok 1= required parameter sid missing 2= required parameter promo missing 3= required parameter username missing 9= invalid username 11= promotion not valid for this customer 100=session expired

Example:

```
<?xml version="1.0" encoding="utf-8" ?>
<CheckPromoCode>
  <Status>0</Status>
  <PromoDescription>
    21 free 11x15cm prints plus free shipping.
  </PromoDescription>
</CheckPromoCode>
```

2.18 CalcOrder

Calculates the prices for given products.

`http://<HTTPServer><HTTPPath>CalcOrder.cpl`

Note: HTTP POST method must be used for parameter passing.

Parameters:

Name	Description
sid	session id
promo	Promotion Code (optional)
did	Delivery id (optional)
username	required if promo is given

For every image we need:

pid1 ... pidN	Product id
qty1 ... qtyN	Number of copies for this image
count	Number of product-quantity pairs in query

Return values:

Name	Description
Currency	3 character currency code
Total	Total price
VatText	Vat description

Name	Description
VatPercent	Vat percent (divide by 100)
VatAmount	Vat amount
DeliveryPrice	Delivery price
DeliveryName	Name of the delivery method
PromoDiscount	Discount amount given by promotion code
PromoDescription	Description of the promotion offer
Status	0=ok 1= required parameter sid is missing 2= required parameter pid1 is missing 3= required parameter qty1 is missing 4= required parameter count is missing 32= error reading currency 33= error reading delivery data 34= error reading prices 35= error reading product names 100=session expired

For every product row:

Pid	Product id
Name	Product name
Price	Unit price of the product
Qty	quantity of products

Example:

```
<?xml version="1.0" encoding="utf-8" ?>
<CalcOrder>
  <Status>0</Status>
  <Currency>EUR</Currency>
  <VatText>Vat included</VatText>
  <VatPercent>750</VatPercent>
  <VatAmount>228</VatAmount>
  <DeliveryName>Priority Mail</DeliveryName>
  <DeliveryPrice>249</DeliveryPrice>
  <Total>859</Total>
  <Row>
    <Pid>110</Pid>
    <Name>10 x 15 cm</Name>
    <Price>120</Price>
    <Qty>2</Qty>
  </Row>
  <Row>
    <Pid>110</Pid>
    <Name>11 x 15 cm (vga)</Name>
    <Price>120</Price>
    <Qty>1</Qty>
  </Row>
</CalcOrder>
```

2.19 RefreshSession

RefreshSession will update session timeout. After this call session will be alive for another default timeout period (currently 1800 seconds). If optional parameter force is set, the

session will be revived even if it is already expired. Session data is totally removed after certain period of inactivity (normally 24 hours), after that it is impossible to revive session even with force parameter.

`http://<HTTPServer><HTTPPath>RefreshSession.cpl?sid=4324abc`

Parameters:

Name	Description
sid	session id
force	If set to 1, forces session refresh (optional)

Return values:

Name	Description
Status	0=ok 1= required parameter sid is missing 100=session not found

Example:

```
<?xml version="1.0" encoding="utf-8" ?>
<RefreshSession>
  <Status>0</Status>
</RefreshSession>
```

2.20 StartOrder

Starts an order and should be sent before first AddOrderItem query. If StartOrder is called more than once during the session the last call overrides the previous call and all previously entered order items are deleted. Application should use returned orderid in all following queries.

`http://<HTTPServer><HTTPPath>StartOrder.cpl?sid=32ad &promo=megaoffer`

Parameters:

Name	Description
sid	session id
username	username
promo	promotion code (optional)
did	Delivery method id from GetDeliveryMethods (optional)
payid	Payment id from GetPaymentMethods (optional)

Return values:

Name	Description
OrderId	Order id of the new order

Name	Description
PromoDescription	Promo description if promo has given
Status	0=ok 1= required parameter sid missing 2= required parameter username missing 3= invalid did 4= invalid payid 11= username does not match 21= error reading currency 22= error reading default delivery 31= DB error creating order 32= DB error updating session 50= Error reading new order id 100=session not found

Example:

```
<?xml version="1.0" encoding="utf-8" ?>
<StartOrder>
  <Status>0</Status>
  <OrderId>12345</OrderId>
</StartOrder>
```

2.21 AddOrderAddress

If the order is to be delivered to an address other than the customers own address, a new address entry should be entered. Each new call to this query will overwrite previous one.

`http://<HTTPServer><HTTPPath>AddOrderAddress.cpl?sid=32ad &first=...`

Parameters:

Name	Description
sid	Session id
orderid	order id from StartOrder
first	First name
last	Last name
addr1	Address line 1
addr2	Address line 2 (optional)
city	City
zip	Postal code
cntry	3 characters ISO-3166 country code
state	State or province (optional)

Return values:

Name	Description
Status	0=ok 1= sid is missing 2= first is missing 3= last is missing 4= addr1 is missing 5= city is missing 6= cntry is missing 7=zip is missing 8=order id is missing or invalid 100=session not found

Example:

```
<?xml version="1.0" encoding="utf-8" ?>
<AddOrderAddress>
  <Status>0</Status>
  <orderId>555333</orderId>
</AddOrderAddress>
```

2.22 AddOrderItems

Adds all order items into the order. Parameters are the same as for CalcOrder except that orderid is now required. Returns order summary if calc parameter is set. Order rows can be added row by row or all rows in the same time. If there is a need to remove order items from the order then ClearOrder should be used. If no order items are given, call will return order summary information for the items already inserted to the order.

HTTP POST method must be used.

`http://<HTTPServer><HTTPPath>AddOrderItems.cpl`

Parameters:

Name	Description
sid	session id
orderid	order id from StartOrder
did	Delivery method id from GetDeliveryMethods (optional)
payid	Payment id from GetPaymentMethods (optional)
calc	if calc=1 product summary will be returned (optional)
promo	Promotion code (optional)
count	Number of product-quantity pairs in the query

For every image we need:

pid1 ... pidN	Product id
qty1 ... qtyN	Number of copies for this image
img1 ... imgN	Image identifier. Generated by the client or obtained by GetOrderImages query in case of web application.

pid1 ... pidN	Product id
opt1 ... optN	Use color correction for this image (optional) 0= no correction 1= automatic correction [default]
qi1 ... qiN	Quality indicator, in scale of 0-100. (optional)
dname1 ... dnameN	Original file name. (optional)
origx1 ... origxN	Original image width. (optional)
origy1 ... origyN	Original image height. (optional)
origb1 ... origbN	Original image file size. (optional)
wb1 ... wbN	White borders were selected instead of cropping. (optional)
vw1 ... vwN	Image is using variable width. (optional)

Return values:

Name	Description
Currency	3 character currency code
VatAmount	VAT amount amount
VatText	Vat description
VatPercent	Vat percent
DeliveryName	Delivery name
DeliveryPrice	Delivery price in minor units
PromoDiscount	Discount amount given by promotion code
PromoDescription	Promotion description
Total	Total amount
Pid	Product id
Name	Product name
Price	Unit price of the product
Qty	quantity of products
Status	0=ok 1= required parameter sid is missing 2= required parameter orderid is missing 11= order not started 18= invalid order id 32= error reading currency 33= error reading delivery data 34= error reading product prices 35= error reading product names 36= db error adding order row 38= db error updating order 40= empty product id parameter 41= empty imgid parameter 100=session expired

Example:

```
<?xml version="1.0" encoding="utf-8" ?>
<AddOrderItems>
  <Status>0</Status>
```

```

    <Currency>EUR</Currency>
    <VatText>Vat included</VatText>
    <VatPercent>750</VatPercent>
    <VatAmount>228</VatAmount>
    <DeliveryName>Priority Mail</DeliveryName>
    <DeliveryPrice>249</DeliveryPrice>
    <Total>859</Total>
    <Row>
      <Pid>110</Pid>
      <Name>10 x 15 cm</Name>
      <Price>120</Price>
      <Qty>2</Qty>
    </Row>
    <Row>
      <Pid>110</Pid>
      <Name>11 x 15 cm (vga)</Name>
      <Price>120</Price>
      <Qty>1</Qty>
    </Row>
  </AddOrderItems>

```

2.23 ClearOrder

Empties product data that has been entered into order.

`http://<HTTPServer><HTTPPath>ClearOrder.cpl`

Parameters:

Name	Description
sid	session id
orderid	order id from StartOrder

Return values:

Name	Description
OrderId	Order id of the order that was cleared
Status	0=ok 1= required parameter sid missing 2= required parameter orderid missing 15= invalid order id 30= db error clearing order data 100=session expired

Example:

```

<?xml version="1.0" encoding="utf-8" ?>
<ClearOrder>
  <Status>0</Status>
  <OrderId>123456</OrderId>
</ClearOrder>

```

2.24 VerifyCC

Verifies and authorizes the credit card payment. This query must be made thru HTTPS. All stored information (=result code from bank) is cleared after transaction is over or session expires. No credit card info is stored in OIPS at anytime.

`https://<HTTPServer><HTTPPath>VerifyCC.cpl?sid=32ad639&ccexp=0605&ccnbr...`

Parameters:

Name	Description
sid	session id
orderid	order id from StartOrder
ccnbr	Credit card number, only numeric data (0-9)
ccexp	credit card expiration date in format yymm
ccholder	Credit card holder name

Return values:

Name	Description
Status	0=ok 1= sid is missing 2= orderid is missing 3= ccnbr is missing 4= ccexp is missing or invalid 5= amount is missing 6= currency is missing 7= ccholder is missing 40= card rejected 50= transaction failed, try again later 100=session not found

Example:

```
<?xml version="1.0" encoding="utf-8" ?>
<VerifyCC>
  <Status>40</Status>
</VerifyCC>
```

2.25 SetDeliveryMethod

Sets the delivery method to be used.

`http://<OIPSaddress>SetDeliveryMethod.cpl?sid=32ad&orderid=1234&did=1`

Parameters:

Name	Description
sid	session id
orderid	order id from StartOrder

Name	Description
did	valid delivery method id

Return values:

Name	Description
OrderId	Order id of the order that was updated
Status	0=ok 1= required parameter sid missing 2= required parameter orderid missing 3= required parameter did missing 10= invalid orderid 11= invalid delivery id 30= error updating session 100=session not found

Example:

```
<?xml version="1.0" encoding="utf-8" ?>
<SetDeliveryMethod>
  <Status>0</Status>
  <OrderId>1234</OrderId>
</SetDeliveryMethod>
```

2.26 SetPaymentMethod

Sets the payment method to be used. When “VerifyCC” query is called, then the payment method is automatically set to the corresponding credit card given via this query.

`http://<OIPSaddress>SetPaymentMethod.cpl?sid=32ad43c&orderid=1234&payid=4`

Parameters:

Name	Description
sid	session id
orderid	order id from StartOrder
payid	valid payment method id

Return values:

Name	Description
OrderId	Order id of the order that was updated
Status	0=ok 1= required parameter sid missing 2= required parameter orderid missing 3= required parameter payid missing 10= invalid orderid 11= invalid payment method 30= db error updating order 100=session not found

Example:

```
<?xml version="1.0" encoding="utf-8" ?>
<SetPaymentMethod>
  <Status>0</Status>
  <OrderId>1234</OrderId>
</SetPaymentMethod>
```

2.27 ConfirmOrder

Order is marked ready for collection, temporarily upload data is removed, and the current OIPS session is destroyed. After that you cannot modify this order or make any queries within this session id.

`http://<HTTPServer><HTTPPath>ConfirmOrder.cpl?sid=xxx&orderid=1234`

Note: Current version will also automatically send a confirmation e-mail to the customer.

Parameters:

Name	Description
sid	session id
orderid	order id from StartOrder

Return values:

Name	Description
OrderId	Order id of the order that was confirmed
Status	0=ok 1= required parameter sid missing 2= required parameter orderid missing 11= invalid orderid 12= payment method is not set 14= Order contains no products 15= Payment method is Credit card, but clearing is not done 16= Current total amount exceeds clearing amount 17= All images are not uploaded 31= db error updating order 100=session expired

Example:

```
<?xml version="1.0" encoding="utf-8" ?>
<ConfirmOrder>
  <Status>0</Status>
  <OrderId>1234</OrderId>
</ConfirmOrder>
```

2.28 CancelOrder

If something unexpected happens after “StartOrder” or customer wants to cancel order, CancelOrder can be used to clear off the order and reverse the possible credit card authorisation.

`http://<HTTPServer><HTTPPath>CancelOrder.cpl?sid=xxx&orderid=4321`

Parameters:

Name	Description
sid	Session id
orderid	Order id from StartOrder

Return values:

Name	Description
OrderId	Order id of the cancelled order
Status	0=ok 1= required parameter sid missing 2= required parameter orderid missing 11= invalid orderid 30= db error cancelling order 31= db error updating session 100=session expired

Example:

```
<?xml version="1.0" encoding="utf-8" ?>
<CancelOrder>
  <Status>0</Status>
  <OrderId>4321</OrderId>
</CancelOrder>
```

2.29 GetImageServer

Image server host name is returned by SessionLogin, but can be explicitly inquired by “GetImageServer” call.

`http://<HTTPServer><HTTPPath>GetImageServer.cpl?sid=xxx&oemid`

Parameters:

Name	Description
sid	session id
oemid	OEM id provided by Colorplaza

Return values:

Name	Description
ImageServer	Image server address and path

Name	Description
Status	0=ok 1= required parameter sid missing 2= required parameter oemid missing 15= image server not defined. OIPS configuration error 100=session expired

Example:

```
<?xml version="1.0" encoding="utf-8" ?>
<GetImageServer>
  <Status>0</Status>
  <ImageServer>imgsrv1.colormailer.com/path/to/upload/</ImageServer>
</GetImageServer>
```

3. Image Server

Image Server is separated cluster of machines and not part of OIPS as such. Address of Image Server can be found from return value of “SessionLogin” or querying “GetImageServer”.

Image Server exposes interfaces to: upload images, view images in full size or view images in predefined size (at the moment thumbnail).

Image Server and OIPS have separated communication layer between them and it is not covered in this document.

3.1 Upload Image(s)

Uploading images should be sent using multipart form data as defined in RFC 1867 (<http://www.ietf.org/rfc/rfc1867.txt>). Sender should send the images one by one. Currently only one image per upload is supported.

RFC 1867 form data should be send into following URL:

```
<ImageServer><parameter list>
```

Above is because some of the applications, that currently use upload system, are not able to send form parameters at the beginning of the form we need to add parameters in the URL.

3.1.1 Parameters

All parameters should put in the URL not into form. Few special cases should be handled in the following way:

Separator:	parameter separator is '+' character.
Empty parameter:	should be marked with '-' character.
'?' in redirect and refresh:	instead of '?' character '@' should be used.
'&' in redirect and refresh:	Instead of '&' characters '@@' should be used.

Whole URL with all parameters is:

```
<ImageServer><client>+<sid>+<oips sid>+<#of imgs>+<redirect url wo/
http://>+<refresh url wo/ http://>+<rotate>+<lang>+<xml output>+<image id>.xup
```

Please note the extension at the end of the URL.

URL Parameters:

Name	Description
<client>	Client type. Based on this some special handling is done. CLI=Off line client HTML=HTML client JAVA=Java client AX=ActiveX client
<sid>	Session id of session located on application server. For online applications. Leave it to '-' if unsure.

Name	Description
<oips sid>	OIPS session id from SessionLogin.
<#of imgs>	Number of images in this transmission. Should be 1.
<redirect url>	Where should the sender be redirected after upload is done. For online applications. Note: if this is used then all images should be send in one form, or otherwise we cannot guarantee the correctness of the function. Note2: URL should be given without 'http://'.
<refresh url>	This URL is queried during the upload session. For online applications. Note: for example URL that Image Server should refresh in order to keep Session live on other server. Note2: URL should be given without 'http://'.
<rotate>	If images should be automatically rotated based on the information read from EXIF of the image. For online applications. Leave it to '-' if unsure.
<lang>	2 char ISO-639 language code. Certain client types require feedback during upload session. This defines the language to be used. For online applications. Leave it to '-' if unsure.
<xml output>	Values: yes/no At the end of the image transmission a summary XML reply could be sent back. Leave it to '-' if unsure.
<image id>	Unique image identifier. The same as given to AddOrderItems.

Example (all parameters have values):

```
<ImageServer>html+123456+654321+10+www.xxx.ch@jump=now@@or=never+www.xyz.com@para=1@@para2=2+yes+en+no+1234567_abc_1.xup
```

Example (there is no redirect parameter given):

```
<ImageServer>java+123456+654321+10+-+www.xyz.com@para=1@@para2=2+yes+en+no+432.xup
```

Example (typical off line client call):

```
<ImageServer>cli+-+4321+1+-+--+yes+2332195_43.xup
```

3.1.2 RFC 1867 Data Stream

Example of RFC 1867 data stream:

```
Content-type: multipart/form-data, boundary=---7d37b202002da
```

```
---7d37b202002da
```

```
Content-Disposition: form-data; name="file1"; filename="D:\Testikuvat\tipi.jpg"
```

```
Content-Type: image/jpeg
```

```
<DATA>
```

```
---7d37b202002da
```

```
Content-Disposition: form-data; name="file2"; filename="D:\Testikuvat\tipi2.jpg"
```

```
Content-Type: image/jpeg
```

```
<DATA>
```

```
---7d37b202002da--
```

3.1.3 XML Reply

If XML reply is requested then following values will be returned:

Name	Description
Sid	Value that passed as <sid> . Not in use.
Gstring	Value that passed as <oips sid>
Status	<p>0= ok</p> <p>General Server errors: 200=server config error; no config file 201=server error; memory allocation failed 202=server error; memory reallocation failed 220=server error; cannot open dump 221=server error; cannot open debug</p> <p>URI and Parameter errors: 203= Cannot found boundary mark 204= Wrong request method 205= Content length is 0 206= Client type not set 207= <sid> not set 208= <oips sid> not set 209= <#of images> not set 222= malformed uri</p> <p>Config file errors: 210-219= config file errors</p>

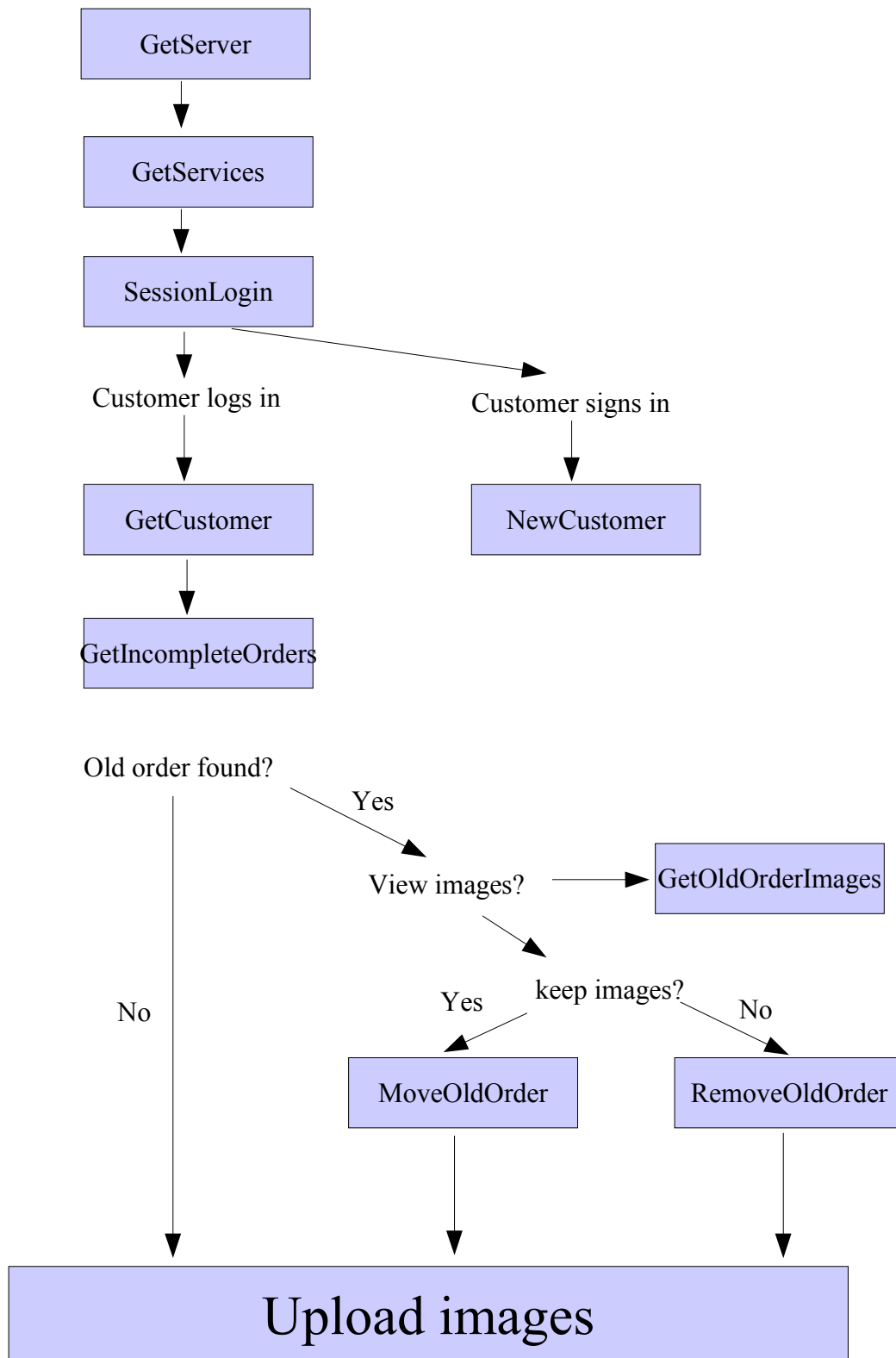
For each Image:

Name	Description
Nbr	Sequence # of the image in the upload session.
Param	Parameter name in the form data when uploaded eg. in form name="file1" -> here file1
ByteSize	Size of the image.
XSize	X in pixels.
YSize	Y in pixels.
TransferStatus	<p>Upload status 0= ok</p> <p>Image errors: 300=image not jpeg 301=unsupported jpeg format</p> <p>See status codes above(201-209). If multiple files are uploaded in the same stream then this can vary image by image.</p>

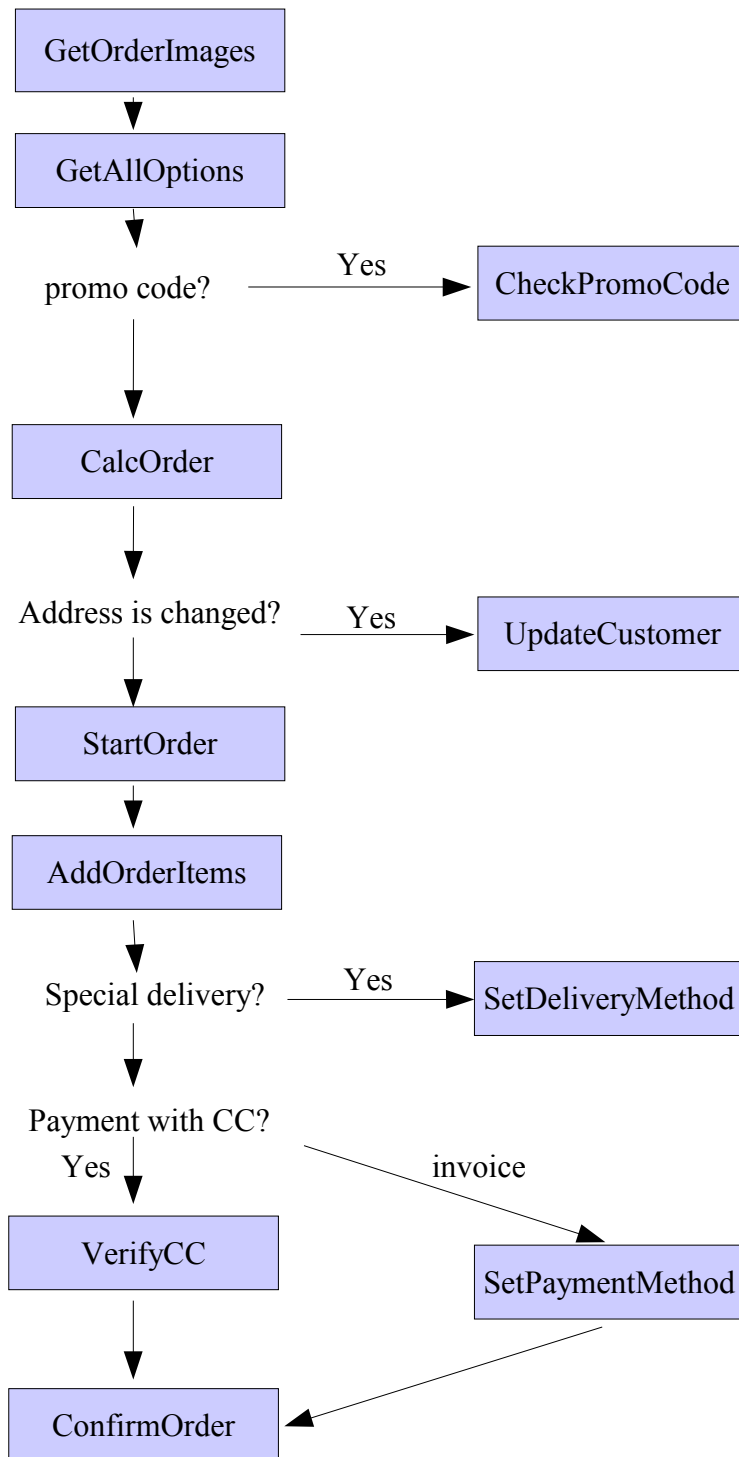
Example:

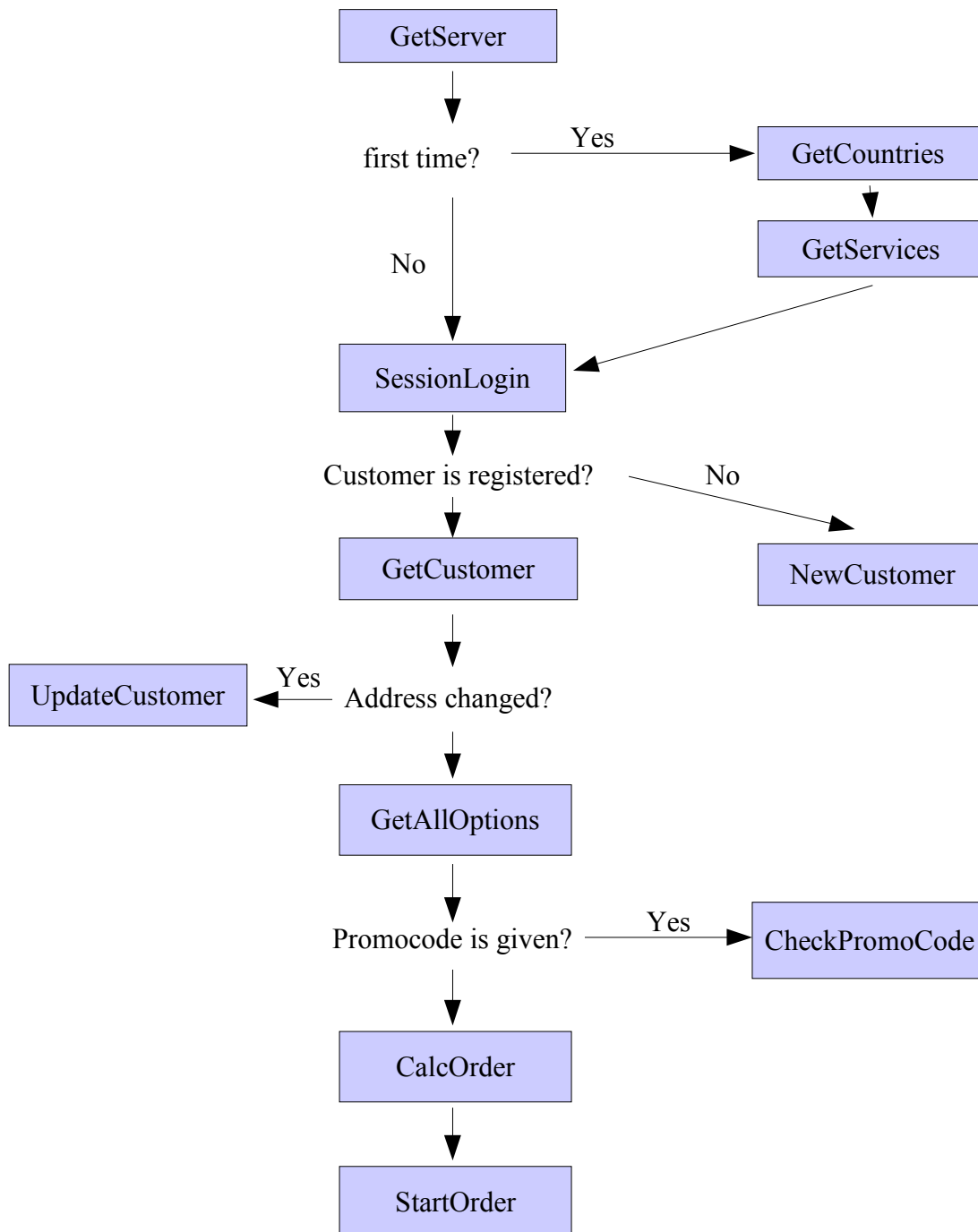
```
<?xml version="1.0" encoding="utf-8" ?>
<ImageUpload>
  <Status>0</Status>
  <Sid>123456</Sid>
  <Gstring>654321</Gstring>
  <Images>
```

```
        <Image>
          <Nbr>1</Nbr>
          <ByteSize>1200</ByteSize>
          <XSize>640</XSize>
          <YSize>480</YSize>
          <TransferStatus>0</TransferStatus>
          <ImageServerStatus>0</ImageServerStatus>
        </Image>
      </Images>
    </ImageUpload>
```

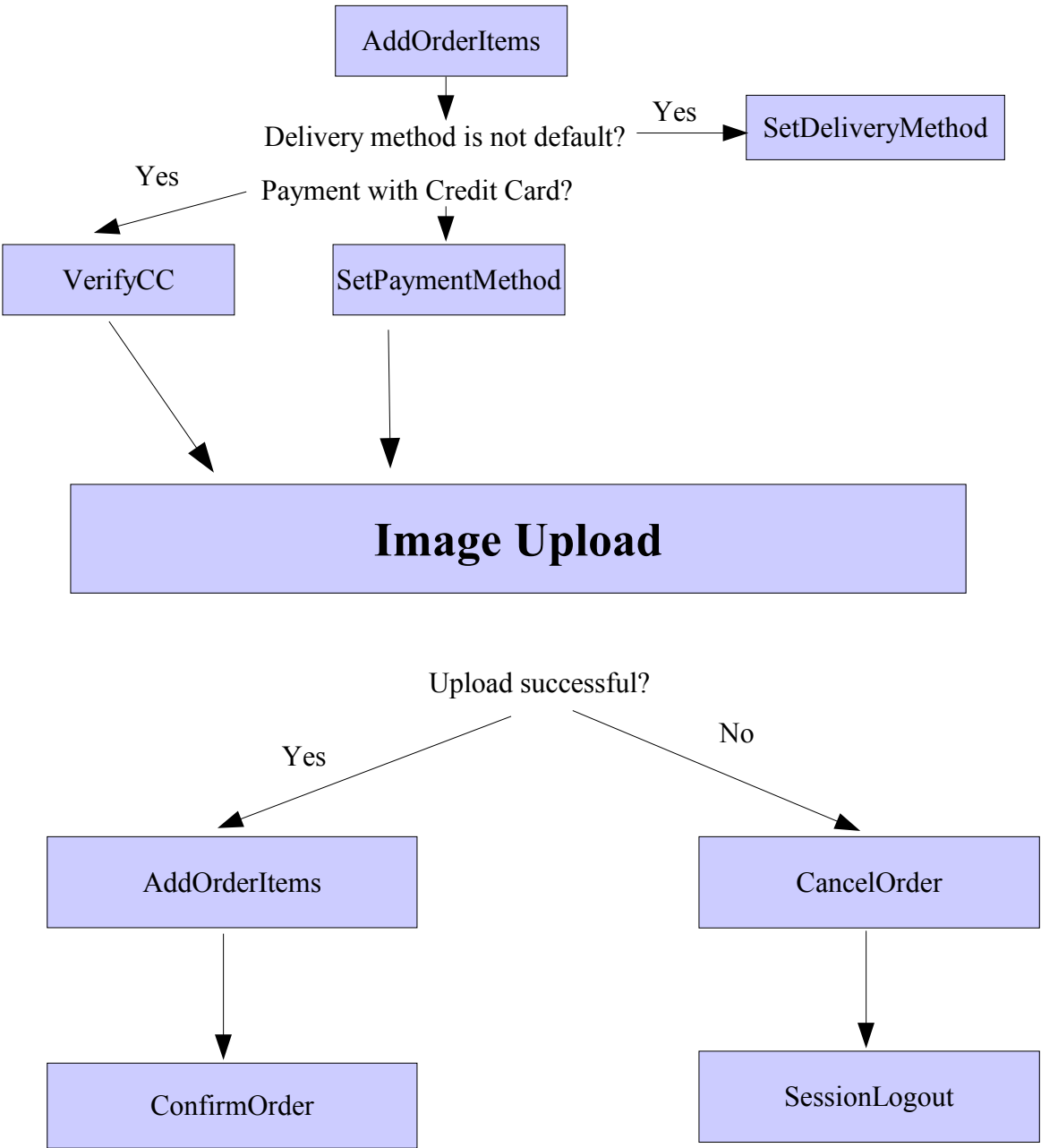
Appendix A - Order flow example for webupload

Webupload order flow after upload.



Appendix B - Order flow example for client sw

Client flow after **StartOrder**



Appendix C – OIPS API License

OIPS API LICENSE AGREEMENT

This is a legal agreement between You and Colorplaza SA ("Copla").

YOU MUST READ AND AGREE TO THE TERMS OF THIS SOFTWARE LICENSE AGREEMENT BEFORE THE OIPS API CAN BE USED. BY USING THE OIPS API, YOU ARE AGREEING TO BE BOUND BY THE TERMS AND CONDITIONS OF THIS SOFTWARE LICENSE AGREEMENT. IF YOU DO NOT AGREE WITH THE TERMS AND CONDITIONS OF THIS SOFTWARE LICENSE AGREEMENT, YOU SHOULD NOT USE THE OIPS API. BY DOING SO, YOU FOREGO ANY IMPLIED OR STATED RIGHTS TO USE THE OIPS API. IF YOU ARE AGREEING TO THIS LICENSE ON BEHALF OF A COMPANY YOU REPRESENT THAT YOU ARE AUTHORIZED TO BIND THE COMPANY TO THIS AGREEMENT

SOFTWARE LICENSE AGREEMENT

Copla grants You a limited, non-exclusive, nontransferable license, under Copla Copyrights, use the OIPS API and any revisions thereto made available by Copla according to the terms set forth below.

DEFINITIONS

For the purpose of this Agreement, the following terms shall have the following meanings:

"OIPS API" means the computer software programs and Application Program Interfaces (API) that implement Ordering Interface for Printing System (OIPS), and associated documentation, that may be delivered by Copla

"You(r)" means the company, entity or individual who submits the request to use OIPS API, or on whose behalf such request is submitted.

"Copla Copyrights" means Copla's copyrights in and to the OIPS API.

LICENSE

You may:

(a) use the OIPS API to evaluate, test and develop printing interface from Your client or server platform to send images to be printed by Copla;

You may not, and may not permit others to:

(a) use the OIPS API or its documentation for the purposes of development or commercialization of any software, product, tools, intellectual property or services competitive with the current products or services offered by Copla or which might reasonably be expected to be offered by Copla or for any purpose other than internal use except as permitted above;

(b) create derivative works of or otherwise alter, adapt, modify, duplicate or translate the OIPS API in any way for any purpose except the purposes expressly permitted above;

(d) distribute, deploy, provide copies of, rent, loan, lease, assign, transfer, or grant any rights in the OIPS API or derivative works thereof in any form to any person.

This license is not a sale. Title, Copla Copyrights and all other rights to the OIPS API remain with Copla.

COMMERCIALIZATION

The OIPS API is protected by Copla Copyrights. Copla hereby reserves all rights in and to Copla Copyrights and no license is granted under Copla Copyrights in this Software License Agreement. Copla generally licenses Copla Copyrights for commercialization pursuant to the terms of either Copla's Standard Software Source Code License Agreement or Copla's Standard Product License Agreement. A copy of Copla's either License Agreement can be obtained on request by email from: oips@colorplaza.com

TERMINATION

This license will continue until terminated. Unauthorized use, reproduction or distribution of the OIPS API or any other failure to comply with the above terms will result in automatic termination of this Agreement and will make available to Copla other legal remedies.

WARRANTY DISCLAIMER

THE EXAMPLE CODE IS PROVIDED ON AN "AS IS" BASIS. Copla AND ITS SUPPLIERS MAKE AND YOU RECEIVE NO OTHER WARRANTIES OR CONDITIONS, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE

OR IN ANY COMMUNICATION WITH YOU, AND Copla AND ITS SUPPLIERS SPECIFICALLY DISCLAIM ANY IMPLIED WARRANTY OF MERCHANTABILITY, SATISFACTORY QUALITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT AND THEIR EQUIVALENTS. Copla does not warrant that the OIPS API will (i) comply in all respects with published protocol descriptions, (ii) be error free or (iii) meet Your specific requirements. SOME STATES OR OTHER JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO THE ABOVE EXCLUSIONS MAY NOT APPLY TO YOU. YOU MAY ALSO HAVE OTHER RIGHTS THAT VARY FROM STATE TO STATE AND JURISDICTION TO JURISDICTION.

LIMITATION OF LIABILITY

LIMITATION OF LIABILITY IN NO EVENT WILL Copla OR ITS SUPPLIERS BE LIABLE FOR ANY DAMAGES, INCLUDING BUT NOT LIMITED TO LOSS OF OR CORRUPTION TO DATA, LOST PROFITS, LOSS OF CONTRACTS, COST OF PROCUREMENT OF SUBSTITUTE PRODUCTS OR ANY OTHER DIRECT, SPECIAL, INCIDENTAL, PUNITIVE, CONSEQUENTIAL OR INDIRECT DAMAGES, LOSSES, COSTS OR EXPENSES OF ANY KIND ARISING FROM THE SUPPLY OR USE OF THE OIPS API, HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY (INCLUDING WITHOUT LIMITATION NEGLIGENCE). THIS LIMITATION WILL APPLY EVEN IF Copla OR AN AUTHORIZED DISTRIBUTOR HAS BEEN ADVISED OF THEIR POSSIBILITY. THIS LIMITATION WILL NOT APPLY TO THE EXTENT THAT LIABILITY MAY NOT BY LAW BE LIMITED OR EXCLUDED. SOME STATES OR OTHER JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATIONS AND EXCLUSIONS MAY NOT APPLY TO YOU.

LANGUAGE

The parties hereto confirm that it is their wish that this Agreement, as well as other documents relating hereto, have been and shall be written in the English language only.

GENERAL

This Agreement is governed by the laws of Switzerland and the intellectual property laws of Switzerland. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation. This Agreement is the entire agreement between us and supersedes any other communications, representations or advertising with respect to the OIPS API. The waiver of a breach of any term hereof shall in no way be construed as a waiver of any other term or breach hereof. If any provision of this Agreement is held invalid or unenforceable, such provision shall be revised to the extent necessary to cure the invalidity or unenforceability, and the remainder of the Agreement shall continue in full force and effect. The person entering into this Software License Agreement represents and warrants that such person is authorized to bind and does so bind his or her principal or employer and has adequate legal capacity to enter this Software License Agreement. You may not assign this Software License Agreement or any rights or obligations hereunder, directly or indirectly, by operation of law or otherwise, without the prior written consent of Copla. Subject to the foregoing sentence, this Software License Agreement shall inure to the benefit of and be binding upon the parties and their respective successors and permitted assigns.

Appendix D – Reference implementation

Reference implementation is available per request. Implementation covers all OIPS functions and is using off line client workflow.

Reference implementation is made using VC++ and it is called “OIPsv2 Client Object Model”.

Both binary and sources are included in the reference implementation package and the package is covered by the following license:

REFERENCE IMPLEMENTATION LICENSE AGREEMENT

This is a legal agreement between You and Colorplaza SA (“Copla”).

YOU MUST READ AND AGREE TO THE TERMS OF THIS SOFTWARE LICENSE AGREEMENT BEFORE THE EXAMPLE CODE CAN BE USED OR INSTALLED. BY INSTALLING THE EXAMPLE CODE, OR USING THE EXAMPLE CODE, YOU ARE AGREEING TO BE BOUND BY THE TERMS AND CONDITIONS OF THIS SOFTWARE LICENSE AGREEMENT. IF YOU DO NOT AGREE WITH THE TERMS AND CONDITIONS OF THIS SOFTWARE LICENSE AGREEMENT, YOU SHOULD NOT INSTALL OR USE THE EXAMPLE CODE. BY DOING SO, YOU FOREGO ANY IMPLIED OR STATED RIGHTS TO INSTALL OR USE THE EXAMPLE CODE. IF YOU ARE AGREEING TO THIS LICENSE ON BEHALF OF A COMPANY YOU REPRESENT THAT YOU ARE AUTHORIZED TO BIND THE COMPANY TO THIS AGREEMENT

SOFTWARE LICENSE AGREEMENT

Copla grants You a limited, non-exclusive, nontransferable license, under Copla Copyrights, install and use the Example Code and any revisions thereto made available by Copla according to the terms set forth below.

DEFINITIONS

For the purpose of this Agreement, the following terms shall have the following meanings:

"Example Code" means the computer software programs in source or binary form that implement OIPS interfaces, and associated documentation, that are included in any source code files that may be delivered by Copla

"You(r)" means the company, entity or individual who submits the request to receive Example Code, or on whose behalf such request is submitted.

"Copla Copyrights" means Copla's copyrights in and to the Example Code.

LICENSE

You may:

- (a) use the Example Code for internal evaluation and internal research purposes;
- (b) use the Example code to evaluate, test and verify products that are wholly or partially of your own design;
- (c) create and use derivative works of the Example Code internally solely for the purposes permitted above; and
- (d) reproduce the Example Code and Your derivative works thereof as reasonably necessary for the purposes permitted above and for backup purposes, provided that You reproduce, unaltered, all proprietary notices on or in the copy.

You may not, and may not permit others to:

- (a) use the Example Code for the purposes of development or commercialization of any software, product, tools, intellectual property or services competitive with the current products or services offered by Copla or which might reasonably be expected to be offered by Copla or for any purpose other than internal use except as permitted above;
- (b) reproduce the Example Code except as permitted above;
- (c) create derivative works of or otherwise alter, adapt, modify or translate the Example Code in any way for any purpose except the purposes expressly permitted above;
- (d) distribute, deploy, provide copies of, rent, loan, lease, assign, transfer, or grant any rights in the Example

Code or derivative works thereof in any form to any person.

This license is not a sale. Title, Copla Copyrights and all other rights to the Example Code and any copy made by You remain with Copla.

COMMERCIALIZATION

The Example Code is protected by Copla Copyrights. Copla hereby reserves all rights in and to Copla Copyrights and no license is granted under Copla Copyrights in this Software License Agreement. Copla generally licenses Copla Copyrights for commercialization pursuant to the terms of either Copla's Standard Software Source Code License Agreement or Copla's Standard Product License Agreement. A copy of Copla's either License Agreement can be obtained on request by email from: oips@colorplaza.com

TERMINATION

This license will continue until terminated. Unauthorized use, reproduction or distribution of the Example Code or any other failure to comply with the above terms will result in automatic termination of this Agreement and will make available to Copla other legal remedies. This license will also automatically terminate if You go into liquidation, suffer or make any winding up petition, make an arrangement with Your creditors, or suffer or file any similar action in any jurisdiction in consequence of debt. Upon termination of this license for any reason You will destroy all copies of the Example Code and all derivative works thereof. Any use of the Example Code or derivative works thereof after termination is unlawful.

WARRANTY DISCLAIMER

THE EXAMPLE CODE IS PROVIDED ON AN "AS IS" BASIS. Copla AND ITS SUPPLIERS MAKE AND YOU RECEIVE NO OTHER WARRANTIES OR CONDITIONS, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE OR IN ANY COMMUNICATION WITH YOU, AND Copla AND ITS SUPPLIERS SPECIFICALLY DISCLAIM ANY IMPLIED WARRANTY OF MERCHANTABILITY, SATISFACTORY QUALITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT AND THEIR EQUIVALENTS. Copla does not warrant that the Example Code will (i) comply in all respects with published protocol descriptions, (ii) be error free or (iii) meet Your specific requirements. SOME STATES OR OTHER JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO THE ABOVE EXCLUSIONS MAY NOT APPLY TO YOU. YOU MAY ALSO HAVE OTHER RIGHTS THAT VARY FROM STATE TO STATE AND JURISDICTION TO JURISDICTION.

LIMITATION OF LIABILITY

LIMITATION OF LIABILITY IN NO EVENT WILL Copla OR ITS SUPPLIERS BE LIABLE FOR ANY DAMAGES, INCLUDING BUT NOT LIMITED TO LOSS OF OR CORRUPTION TO DATA, LOST PROFITS, LOSS OF CONTRACTS, COST OF PROCUREMENT OF SUBSTITUTE PRODUCTS OR ANY OTHER DIRECT, SPECIAL, INCIDENTAL, PUNITIVE, CONSEQUENTIAL OR INDIRECT DAMAGES, LOSSES, COSTS OR EXPENSES OF ANY KIND ARISING FROM THE SUPPLY OR USE OF THE EXAMPLE CODE, HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY (INCLUDING WITHOUT LIMITATION NEGLIGENCE). THIS LIMITATION WILL APPLY EVEN IF Copla OR AN AUTHORIZED DISTRIBUTOR HAS BEEN ADVISED OF THEIR POSSIBILITY. THIS LIMITATION WILL NOT APPLY TO THE EXTENT THAT LIABILITY MAY NOT BY LAW BE LIMITED OR EXCLUDED. SOME STATES OR OTHER JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATIONS AND EXCLUSIONS MAY NOT APPLY TO YOU.

LANGUAGE

The parties hereto confirm that it is their wish that this Agreement, as well as other documents relating hereto, have been and shall be written in the English language only.

SUPPORT

You acknowledge that Copla offers no support or maintenance of the Example Code.

GENERAL

This Agreement is governed by the laws of Switzerland and the intellectual property laws of Switzerland. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation. This Agreement is the entire agreement between us and supersedes any other communications, representations or advertising with respect to the Example Code. The waiver of a breach of any term hereof shall in no way be construed as a waiver of any other term or breach hereof. If any provision of this Agreement is held invalid or unenforceable, such provision shall be revised to the extent necessary to cure the invalidity or unenforceability, and the remainder of the Agreement shall continue in full force and effect. The person entering into this Software License Agreement represents and warrants that such person is authorized to bind and does so bind his or her principal or employer and has adequate legal capacity to enter this Software License Agreement. You may not assign this Software License Agreement or any rights or obligations hereunder, directly or indirectly, by operation of law or otherwise,

without the prior written consent of Copla. Subject to the foregoing sentence, this Software License Agreement shall inure to the benefit of and be binding upon the parties and their respective successors and permitted assigns.