



KODAK Image Access Client SDK Version 2.0 Frequently Asked Questions

KODAK Image Access Client SDK Questions

Q. What's new in Version 2.0 of KODAK Image Access Client?

- A. Included in the KODAK Image Access Client SDK Version 2.0 are several additional API calls: GetAvailableOutputs, GetTimeToComplete, and ValidateFulfillmentSystemUser. The client can now request from the DLS server information about what type of output products are available, such as PictureCD and Archiving. The client application can request the anticipated time for a particular order to be completed. In addition, the client application can submit information such as name, password, address, phone number, and email address for validation by the DLS server.

Protection for confidential user information can be critical on the Internet. The KODAK Image Access Client SDK now supports secure encrypted communications using Secure Socket Layer (SSL) technology.

Q. Is KODAK Image Access Client SDK Version 2.0 backward compatible with Version 1.0? If not, what should I do?

- A. No, it is not. To take advantage of new features in KODAK Image Access Client 2.0 you need to upgrade to KODAK Enabling Software Version 2.0 and KODAK Image access Client SDK Version 2.0.

Q. Which software development languages can be used with the KODAK Image Access Client SDK?

- A. The KODAK Image Access Client API supports JAVA, C and C++ for development of client applications.

Q. When will you have MACINTOSH Image Access Client SDKs?

- A. The timeframe for support for MACINTOSH clients is undetermined at this time.

Q. I developed a DLS client software application using OPCCLI. How compatible is this with the new KODAK Image Access Client SDK Version 2.0?

A. The KODAK Image Access Client is compatible with your DLS Client software using OPCCLI. To take advantage of features and future compatibility, we recommend migrating your software to use the KODAK Image Access Client SDK libraries.

Q. Why is image count not returned for the GetOrders call? How do I obtain an image count?

A. This feature is not supported by DLS 2.0. To obtain the image count use GetImageRefs(). GetImageRefs() returns the image references in an array. Calculate the image count from the length of the returned image array.

Q. What happens when the client application asks for a product that is not supported?

A. Unpredictable results occur if a requested product is not supported. It is a good practice to request only products returned by the GetAvailableProducts methods.

Q. Why doesn't GetOrders work when I select based on status?

A. The OrderStatus parameter of the GetOrders family of methods is not supported by DLS 2.0. It is best not to specify OrderStatus or OrderID parameters to the GetOrders method. Set these to NOTFOUND and “ ”, respectively. This will be supported in future releases of DLS.

Q. Why is GetOrders not returning any orders?

A. There are several reasons why you might not get any orders back.

- If the order was only recently submitted, it might not have been archived yet. Wait a while for the order to be archived and try again.
- Make sure the CustomerID is not misspelled and that the date range is correct.
- The order may no longer be present on the archive because the DLS software periodically purges orders from the archive automatically.

Q. Why is GetOrdersAdmin not returning any orders?

A. Check the customer ID. GetOrdersAdmin only returns orders with customer IDs in the format of e-mail addresses. The customer ID must contain an @ surrounded by other characters.

Q. What methods in KODAK Image Access Client SDK Version 2.0 are not supported with DLS 2.0?

A. GetTimetoComplete, GetOrderStatus, AreICCProfilesSupported, ValidateFulfillmentSystemUser, and GetAvailableOutputs.

Q. Why is the orderID in the COS file different from the one returned by SubmitCosOrder? When the orderID returned by SubmitCosOrder is passed to GetOrders, why are zero orders returned?

- A. When using the KODAK Image Access Client SDK with DLS 2.0, there are three distinct orderID's:
- The orderID returned by GetNewOrderID() and SubmitCosOrder() is more of a submissionID than an orderID. This ID is only known by the KODAK Image Access Client. It is never known by DLS. It is to be used only for enabling the partial upload feature of the Image Access Client. If the submission failed prior to completion, you may resubmit with the same submissionID to resume the upload from where it left off. This submissionID is also the ID that is passed as an input parameter to GetOrderStatus().
 - The orderID embedded in the COS file appears on the WIP screen and the Archive screen. This DLS WIP orderID is the same as bagID. These orderIDs are not unique; multiple orders with the same orderID may appear on the WIP screen. WIP internally maintains a unique ID to distinguish orders, but this ID is not externally accessible. The WIP orderID is entered by the DLS operator for a scanned order. For a network order, the orderID is the 6-digit orderID embedded within the COS file. The orderID from the COS file appears on the WIP screen. If you submit the same file twice, the same orderID shows up twice on the WIP screen. This creates two separate orders having the same WIP orderID.
The KODAK Image Access Client is completely unaware of this orderID. Only the creator of the COS file and the DLS 2.0 are aware of this ID.
 - The orderID returned by GetOrders() came from the internal Archive database. This is the orderID used by the archive subsystem of DLS to identify orders. It is the same orderID that is returned by the KODAK Image Access Client GetOrders() call. It has no relationship to the previously mentioned submissionID and bagID. You may not view the archive ID of an order on your DLS. It is only kept internally and appears nowhere on the DLS UI. The orderID that appears on the Archive screen is the bagID, not the archiveID.

Q. Why is JAVA required if I am writing a C++ KODAK Image Access Client application?

- A. The KODAK Image Access Client SDK is implemented in JAVA. The native .dll that is supplied provides no functionality other than to delegate each call to the JAVA library.

Enabling Software Questions

Q. I do not see any TOMCAT console window as mentioned in the guides.

- A. Start TOMCAT by navigating to "Start/Programs/Kias/Start Tomcat."
This should open a new DOS console window with an application title
"Catalina."

Q. When I start TOMCAT from the "Start/Programs/Kias/Start Tomcat," a console window opens and immediately closes.

- A. This might happen because JRE 1.3.0 is not installed or is not found on the system. Open a DOS window (WINDOWS) or a terminal window (SOLARIS) and type java-version and verify that the return states java version "1.3.0." If it is not found, install JAVA 1.3.0. Refer to the *KODAK Image Access Client SDK Version 2.0 Developer's Guide* for installation information.

If JRE is installed and TOMCAT still cannot be started, open a new DOS command window and change the directory to c:\tomcat\bin\. Run the *tomcat start* command. This starts TOMCAT in a new window and also shows the environment settings with the classpath that TOMCAT is currently using. This window shows the error message that helps identify the problem.

Q. Is it normal to see a trail of text messages in the TOMCAT console window?

- A. Yes. These messages are the logs from the KODAK Image Access servlet application. The messages might also contain occasional error codes and messages if you encounter any error conditions on the client application. Usually you can ignore these informational messages.

Q. Why do I have to reinstall the KODAK Image Access Enabling Software after reinstalling the KODAK DLS Enhanced Services Software and KODAK DLS System Management Software?

- A. During installation of the KODAK Image Access Enabling Software, common components are upgraded to support the KODAK Image Access Client SDK.