

Kodak Capture Pro Workflow API Documentation

October 2013

For use with Capture Pro Version 4.5 or later

MCE Client SDK & Demo Program

Copyright © 1998-2009 Eastman Kodak Company. All rights reserved.

Contacts: Chris Larson (christopher.larson@kodak.com)

Bruce Rodgers (g.bruce.rodgers@kodak.com)

Remarks

MCE Client SDK is a system development toolkit providing the interfaces to developers who are willing to communicate to Capture Pro Software products which support one or multiple MCE ports. For example, Kodak Capture Pro Software is providing MCE port (1186) for Workflow API, so that developers could control the workflow of the application.

The SDK provides the generic interface for all the different development tools like Visual C++, Visual Basic, etc. This spreadsheet is a demo on how to use programming with MCE Client SDK in Visual Basic (VBA). It also introduces a typical usage of MCE Client SDK.

Demo

STEP 1. Setup Server

Host Address:

(Host Address is a folder on file system where the MCE server is connected. It could be located on local system or it could be on a remote system based on the different products and configurations.)

Port:

(Port is a number for the MCE server virtual port. Typically, one product is with one port, but it is also possible to provide multiple ports for different purpose. It is specified by the different products.)

STEP 2. Prepare Command

Command:

(This is a command which is supported by the MCE server. Details of all supported commands can be found in the "Kodak Capture Pro Software Workflow API" specification document.

This document is subject to change, for the latest version please visit:

www.kodak.com/go/capturepro

The Workflow API is one of the APIs supported by Kodak Capture Pro Software (KC) product family so that a developer can control KC's user interface and workflow via commands. With this API, it enables the integration of KC with customer-developed applications seamlessly.

Basically, the Workflow API contains a set of commands. Other applications can use these commands to control the GUI and workflow of Capture Pro Software. The commands are sent to KC via an MCE protocol which is used by KC. The MCE protocol is a messaging service based on the file system, so before sending MCE messages to KC, it needs to set up the file system so that a connection between your application and KC is established.

IMPORTANT: The Workflow API works differently than the other Capture Pro Software APIs (i.e., the Batch Output Format API and the Indexing API) and it is not based on the API framework.

This document is intending to provide an overview on Kodak Capture Pro Software Workflow API, so that developers can develop an application to control KC with the Workflow API specification. It also describes how to use MCE Client SDK to send Workflow API commands. We expect the reader has product knowledge of KC and understands how the batch structure is defined in the KC product family. The developer for a Workflow API integration should have some basic knowledge of the Windows API, so that they can use the MCE Client SDK to send commands to KC program.

1. MCE Client SDK

1.1 Overview

MCE Client SDK provides a way to send MCE commands to KC so that it can invoke the functionality of Capture Pro Software. Before sending commands to KC, the client application needs to set up some variables so that the SDK can send the commands to the KC Workflow API. These variables include:

- Host Address
- Port

Essentially, the MCE protocol is exchanging information via the file system. Typically, all the MCE services provided on a system will be organized in one root directory on the file system. This folder is called "host address". A different service will be using a different number to identify the service. This number is called "port".

For example, for the Kodak Capture Pro Software product, this host address is typically at "[my documents]\KCS Pro\MCE" where "[my documents]" needs to be interpreted as the absolute file path on the operating system. (e.g., "C:\Documents and Settings\Administrator\My Documents").

The port for the Workflow API is always 1186.

This SDK provides very simple calls to send MCE commands:

- mceSetVariable()
- mceSendCommand()
- mceSendCommandEx()

The MCE protocol fully supports Unicode, so all the commands could be sent in Unicode. But to have the compatibility with an ANSI application, it provides 3 different sets of interfaces to enable Unicode, ANSI and VB (Visual Basic) integration. To differentiate the routine name, we added the following suffix to the routine names:

- W: for Unicode
- A: for ANSI
- VB: for VB

1.2 mceSetVariable()

This routine is used to set up the host address and port before sending an MCE command to Capture Pro Software. To enable the possibility of Capture Pro Software using multiple MCE services or to potentially resolve some multi-thread conflicts, there is a client ID required. For each client ID, it could be setup independently. Typically, use 0 for client ID. The maximum client ID is 127.

Two variable names had been defined:

- HostAddress
- Port

The developer just needs to setup these two variables at beginning of their program once.

Example:

```
BOOL CMainApp::OnInitInstance()
{
    //...

    //setup KC Workflow API
    WORD ret;
    ret = mceSetVariableW(_T("HostAddress"), _T("C:\\Documents and
Settings\\Administrator\\My Documents\\KCS Pro\\MCE"), 0);
    if (ret != MCECLIENTSDK_ERROR_NOERROR) return FALSE;
    ret = mceSetVariableW(_T("Port"), _T("1186"), 0);
    if (ret != MCECLIENTSDK_ERROR_NOERROR) return FALSE;

    //...
}
```

1.3 mceSendCommand()

This routine is used to send a command to Capture Pro Software without needing a response.

Example:

```
BOOL CMainFrame::OnShowKC()
{
    //...

    //show KC
    WORD ret;
    ret = mceSendCommandW(_T("showapp"), NULL, 0);
    if (ret != MCECLIENTSDK_ERROR_NOERROR) return FALSE;

    //move KC window to (0, 0, 400, 300)
    CString strCommand;
    strCommand = _T("x=0");
    strCommand += _T("\r\n");
    strCommand += _T("y=0");
    strCommand += _T("\r\n");
    strCommand += _T("w=400");
    strCommand += _T("\r\n");
    strCommand += _T("h=300");
    ret = mceSendCommandW(_T("movewindow"), strCommand, 0);
    if (ret != MCECLIENTSDK_ERROR_NOERROR) return FALSE;
}
```

```
} //...
```

1.4 mceSendCommandEx()

This routine is used to send a command to Capture Pro Software with a response. Typically, KC will return "OK" message when the command is processed.

1.5 Error Code

The return value of a routine is a number which will indicate the error code. MCECLIENTSDK_ERROR_NOERROR is defined as successful. Error codes had been defined in DFCMCE.h (the include file is contained in the **inc** directory in the **Capture Pro Software Workflow API.zip** file) which should be self-explaining with its macro name.

2. Workflow API

2.1 Overview

There are 4 groups of Workflow API commands that are supported in Capture Pro Software:

- GUI Controlling Commands
- Workflow Controlling Commands
- Function Invoking Commands
- Query Commands

GUI Controlling Commands are used to control the user interface of Capture Pro Software. Workflow Controlling Commands are used to control the basic batch handling workflow. Function Invoking Commands are used to invoke some functions in Capture Pro Software. Query Commands are used to query the status of Capture Pro Software.

Note: Capture Pro Software must be running in order for these commands to be recognized.

2.2 GUI Controlling Commands

2.2.1 Show Application

Show main screen.

Command
showapp

Request (Parameters)
(n/a)

Response
(n/a)

2.2.2 Hide Application

Hide main screen.

Command
hideapp

Request (Parameters)
(n/a)

Response
(n/a)

2.2.3 Move Window

Move main screen of Capture Pro Software to a certain position on the screen.

Command
movewindow

Request (Parameters)
x = <x0>
y = <y0>
w = <width>
h = <height>

Response
(n/a)

2.2.4 Enter Embedded Mode

Hide main screen title bar, so that it will be more like an embedded window.

Command
enterembeddedmode

Request (Parameters)
(n/a)

Response
(n/a)

2.2.5 Leave Embedded Mode

Show main screen title bar as normal.

Command
leaveembeddedmode

Request (Parameters)
(n/a)

Response
(n/a)

2.3 Workflow Controlling Commands

2.3.1 Login

Logout from current user and login with the given name.

Command
login

Request (Parameters)
name = <user name>
password = <password>
bypasspassword = <bypasspassword flag: 1/0>

<password> is optional. When it is not available, it will use blank as password. It will verify if password is correct, if it is incorrect, the normal login dialog will be shown up.

<bypasspassword flag> is optional. When it is set to "1", the password check will be skipped.

Response
(n/a)

2.3.2 New Batch

Create a new batch of a given job.

Command

newbatch

Request (Parameters)

jobname = <job setup name>

batchname = <batch name>

startingdocumentid = <starting document id>

imageaddress = <starting image address; only for Kodak scanners that support image addressing>

counter = <starting counter that will be imprinted; only for Kodak scanners that have imprinters>

Response

(n/a)

2.3.3 Open Batch

Open an existing batch of a given job.

Command

openbatch

Request (Parameters)

jobname = <job setup name>

batchname = <batch name>

Response

(n/a)

2.3.4 Start Scan

Start scanning with a given page setup.

Command

startscan

Request (Parameters)

pagesetupname = <page setup name>

<page setup name> is optional. When it is not given, it will use current selected page setup for scanning.

Response

(n/a)

2.3.5 Stop Scan

Stop scanning.

Command

stopscan

Request (Parameters)

(n/a)

Response

(n/a)

2.3.6 Process Batch

Process the current open batch.

Command

processbatch

Request (Parameters)
(n/a)

Response
(n/a)

2.3.7 Logout

Logout.

Command
logout

Request (Parameters)
(n/a)

Response
(n/a)

2.3.8 Do shortcut.

Start to run a Job Shortcut that has been pre-defined.

Command
doshortcut

Request (Parameters)
shortcutname = <shortcut name>

Response
(n/a)

2.3 Function Invoking Commands

2.3.1 CID Invoking

To simulate a certain menu item to invoke a function in Capture Pro Software.

Command
command

Request (Parameters)
ID = <CID>

<CID> is a named command identity in Capture Pro software

Possible CID values are grouped by the Main Screen menu options:

File Menu

CID_JOBSETUP
CID_PAGESETUP
CID_WORKSTATIONSETUP
CID_IMPORTJOBSETUP
CID_EXPORTJOBSETUP
CID_USERSETUP
CID_CHANGEPASSWORD
CID_LANGUAGE
CID_PRINT
CID_SAVEASPDF
CID_LOGOUT
CID_EXIT
CID_REMOVEFROMTRAY
CID_MINIMIZETOTRAY
CID_BUTTONSETUP

CID_SHORTCUTSETUP
CID_MS
CID_MS_US
CID_MS_METRIC

Batch Menu

CID_NEWBATCH
CID_OPENBATCH
CID_CLOSEBATCH
CID_RENAMEBATCH
CID_REMOVEBLANKIMAGES
CID_OUTPUT
CID_OUTPUTALL
CID_CLEARBATCH
CID_DELETEBATCH
CID_DELETETOENDOFBATCH
CID_CUT
CID_PASTE
CID_BATCHMANAGER
CID_VIEWKCSSVR

View Menu

CID_IMAGEDISPLAY
CID_IMAGEDISPLAY_HIGHQUALITY
CID_IMAGEDISPLAY_HIGHSPEED
CID_IVDISPLAY
CID_IVDISPLAY_1
CID_IVDISPLAY_2
CID_IVDISPLAY_4
CID_IVDISPLAY_8
CID_IVDISPLAY_C
CID_IVFILTER
CID_IVFILTER_FRONT
CID_IVFILTER_BACK
CID_IVFILTER_BLACKANDWHITE
CID_IVFILTER_GRAYSCALE
CID_IVFILTER_COLOR
CID_TVFILTER
CID_TVFILTER_FRONT
CID_TVFILTER_BACK
CID_TVFILTER_BLACKANDWHITE
CID_TVFILTER_GRAYSCALE
CID_TVFILTER_COLOR
CID_TVIMAGESIZESETUP
CID_IMAGEVIEWER
CID_BATCHEXPLORER
CID_THUMBNAILVIEWER
CID_INFORMATION
CID_INDEX
CID_STATUSBAR
CID_TOOLBARS
CID_TOOLBARS_CAPTURE
CID_TOOLBARS_SCANNERADJUSTMENT
CID_TOOLBARS_BATCH

CID_TOOLBARS_VIEW
CID_TOOLBARS_DOCUMENT
CID_TOOLBARS_NAVIGATION
CID_TOOLBARS_FLAG
CID_TOOLBARS_EDIT
CID_TOOLBARS_INDEX
CID_ICONSIZING
CID_ICONSIZING_SMALL
CID_ICONSIZING_MEDIUM
CID_ICONSIZING_LARGE
CID_ICONTEXT
CID_ICONTEXT_NOLABEL
CID_ICONTEXT_LABELONRIGHT
CID_ICONTEXT_LABELBELOW
CID_ZOOM
CID_ZOOM_FITMODE
CID_ZOOM_ACTUALMODE
CID_ZOOM_ZOOMIN
CID_ZOOM_ZOOMOUT
CID_ZOOM_ZOOMSTEPSETUP
CID_PANIMAGESTO
CID_PANIMAGESTO_TOP
CID_PANIMAGESTO_BOTTOM
CID_PANIMAGESTO_LEFT
CID_PANIMAGESTO_RIGHT
CID_PANIMAGESTO_SAMEPOSITION
CID_PANSTEPSETUP
CID_LAYOUT
CID_LAYOUT_CLASSIC
CID_LAYOUT_ENHANCED

Capture Menu

CID_START
CID_STOP
CID_PAGESETUPLIST
CID_SIDESTOCAPTURE
CID_SIDESTOCAPTURE_DEFAULT
CID_SIDESTOCAPTURE_TWOSIDED
CID_SIDESTOCAPTURE_ONESIDEDFRONT
CID_SIDESTOCAPTURE_ONESIDEDBACK
CID_SCANNERADJUSTMENTS
CID_SELECTSCANNER
CID_IMPORTIMAGES
CID_AUTOIMPOPT
CID_AUTOIMPORTSETUP
CID_SETIMAGEADDRESS
CID_SETCOUNTER

Document Menu

CID_NEWDOCUMENT
CID_ATTACHPAGES
CID_INSERTPAGES
CID_RESCANPAGES
CID_BESPLITDOCUMENT

CID_MERGEINTOPREVIOUS
CID_DELETEDOCUMENT
CID_DELETEDOCUMENTRANGE
CID_DELETETOENDOFDOCUMENT
CID_GOTODOCUMENT
CID_GOTODOCUMENT_NUMBER
CID_GOTODOCUMENT_FIRST
CID_GOTODOCUMENT_LAST
CID_GOTODOCUMENT_PREVIOUS
CID_GOTODOCUMENT_NEXT
CID_GOTOIMAGE

Edit Menu

CID_UNDO
CID_COPY
CID_COPYIMAGEAREA
CID_COPYIMAGETOJOBSETUP
CID_SELECTALL
CID_SELECTDOCUMENT
CID_SELECTFILTERED
CID_DESELECTALL
CID_SELECTFLAGGED
CID_DELETESELECTED

Tools Menu

CID_SELECTOR
CID_SELECTIONMODE
CID_SELECTIONMODE_IMAGE
CID_SELECTIONMODE_PAGE
CID_SELECTIONMODE_PAGESIDE
CID_MAGNIFY
CID_PAN
CID_DRAWREGION
CID_MAGNIFICATIONRATIO
CID_ROTATE
CID_ROTATE_90
CID_ROTATE_180
CID_ROTATE_270
CID_FLAGTOOL
CID_FLAGALL
CID_FLAGALLSELECTED
CID_FLAGDOCUMENT
CID_FLAGFILTERED
CID_UNFLAGDOCUMENT
CID_UNFLAGALLSELECTED
CID_UNFLAGALL
CID_SPLITDOCUMENT
CID_IVSPLITDOCUMENT
CID_EXPANDALL
CID_COLLAPSEALL
CID_DELETESELECTED_RC
CID_CROP
CID_BLANK
CID_DESELECTREGION

CID_APPLYREGIONTODOCUMENT
CID_APPLYREGIONTOALLSELECTED
CID_DRAGANDDROPMOVE
CID_IQC

Index Menu

CID_EDITINDEXFIELDS
CID_EDITBATCHFIELDS
CID_EDITDOCUMENTFIELDS
CID_PREVIOUSDOCUMENT
CID_NEXTDOCUMENT
CID_NEXTINVALID
CID_ENTERKEYBEHAVIOR
CID_ENTERKEYBEHAVIOR_NEXTDOCUMENT
CID_ENTERKEYBEHAVIOR_NEXTINVALID
CID_ENTERKEYBEHAVIOR_SAVEANDEXITINDEXING
CID_ZOOMZONES
CID_DRAGANDDROPOCR
CID_OCRLANGUAGE
CID_SAVEANDEXITINDEXING
CID_CANCELANDEXITINDEXING
CID_ENTERKEYACTION
CID_INDEX_PREVITEM
CID_INDEX_NEXTITEM

Help Menu

CID_HELP
CID_REFERENCEGUIDE
CID_ONLINEHELP
CID_TUTORIAL
CID_ENTERLICENSECODE
CID_REGISTER
CID_ONLINE
CID_ABOUT

Others

CID_SHOWAPP
CID_HIDEAPP
CID_CTRL_PAGESETUPLIST
CID_CTRL_INDEXOCRLANGUAGELIST
CID_CTRL_SIDESTOCAPTUREOPTIONS
CID_CTRL_SELECTIONMODEOPTIONS
CID_CTRL_IWDISPLAYOPTIONS
CID_CTRL_SHORTCUTOPTIONS

Scanner Adjustments

CID_BTN_ITHRESHOLD

Response

(n/a)

2.4 Query Commands

2.4.1 Get Current Language

Get current UI language of KC for the current user.

Command
getlanguage

Request (Parameters)
(n/a)

Response
langid = <language id>

2.4.2 Get Current Application State

Get current application state of Capture Pro Software.

Command
getappstate

Request (Parameters)
(n/a)

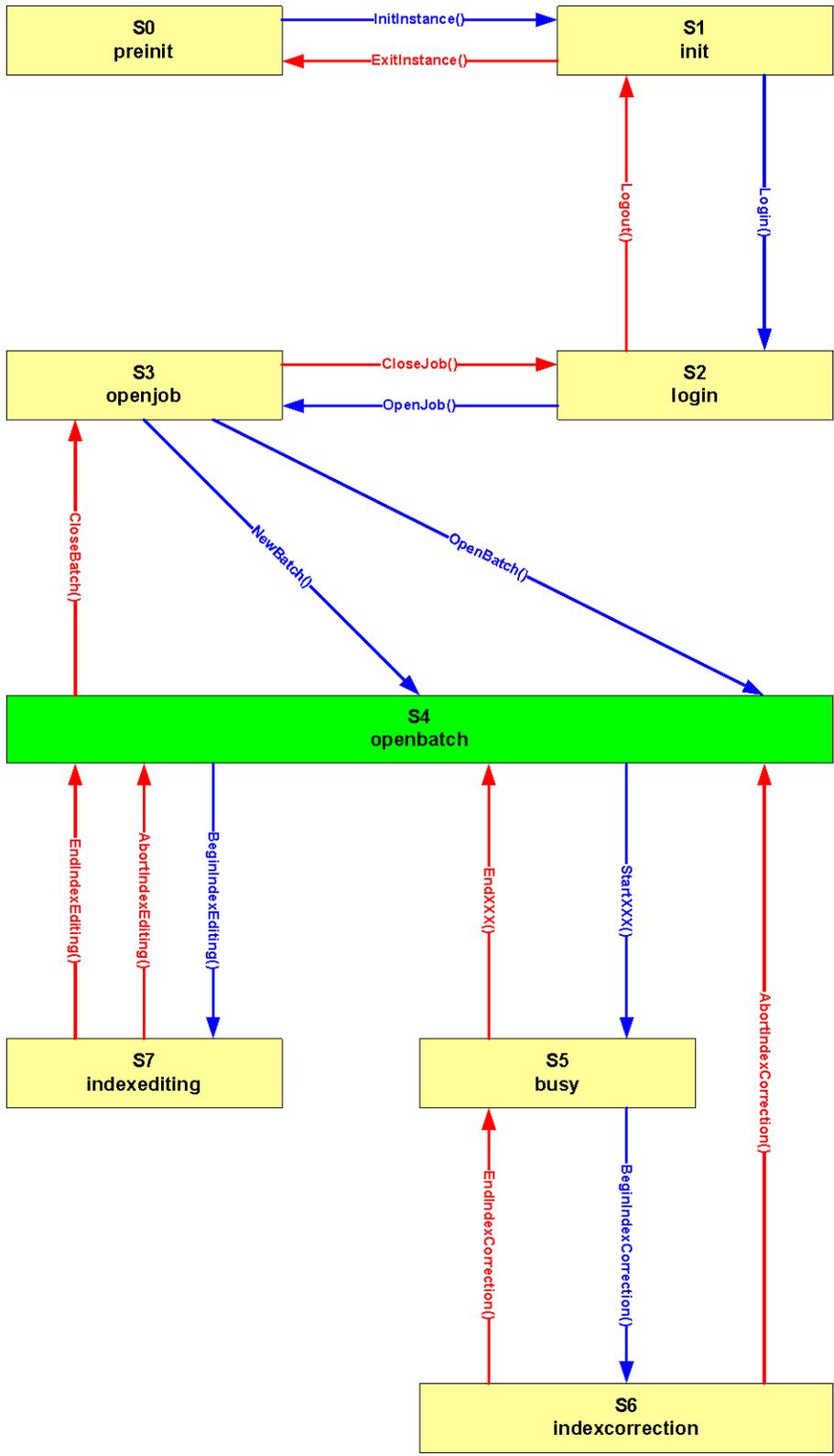
Response
appstate = <app state>

The Application State can be one of the following values:

preinit
init
login
openjob
openbatch
busy
indexcorrection
indexediting

The following chart and diagram describes each state:

	S0 preinit	S1 init	S2 login	S3 openjob	S4 openbatch	S5 busy	S6 indexcorrection	S7 indexediting
Current User ID	No	No	Yes	Yes	Yes	Yes	Yes	Yes
Current User Profile	No	No	Yes	Yes	Yes	Yes	Yes	Yes
Current Job Setup ID	No	No	No	Yes	Yes	Yes	Yes	Yes
Current Job Setup	No	No	No	Yes	Yes	Yes	Yes	Yes
Current Batch	No	No	No	No	Yes	Yes	Yes	Yes
Current Page Setup ID	No	No	No	No	Yes	Yes	Yes	Yes
Current Page Setup	No	No	No	No	No	Yes	Yes	No



3. Sample Code

There is an example available for developers to have better understanding on how to use the MCE Client SDK to send commands to a running instance of Capture Pro Software. It is an Excel spreadsheet which was written in Visual Basic. In the ZIP file entitled **Capture Pro Software Workflow API.zip**, the sample source code can be found in the **samples** sub-directory:

Filename: **DFC MCE Client SDK VBA Tester (Excel 2003).xls** To open it in Excel, please make sure the macro security settings had been correctly setup. After opening up the Excel file, press Alt-F11 to view the Visual Basic source code.

To run the sample code, copy the [DFCMCE.dll](#), [DFCBAS.dll](#), [xerces-c_3_0.dll](#), [msvc71.dll](#), [msvcr71.dll](#) and [mfc71u.dll](#) files from the **bin** directory of the **zip file** to the Windows system directory. For example, [c:\Windows\System32\](#) (Note: For Windows 7, 64-bit systems you must copy the above files to the [c:\Windows\SysWOW64](#) directory).

Open "DFC MCE Client SDK VBA Tester (Excel 2003).xls" and follow the instructions to run the macro.

Version	Date	Author	Changes:
1.0	Jan 2010	CL	Initial Version
1.1	Mar 2010	CL/BR	Updates to configuration
1.2	June 2012	CL	New commands for new features; configuration updates
1.3	Oct 2013	CL	New options for New Batch command; Win 7 64-bit support for sample demo program

Contacts: Chris Larson (christopher.larson@kodak.com)