HP Embedded Capture (HP EC)

API Reference Guide

Version 1.5.0

March 2016

Table of contents

List of tables

List of figures

1 API Introduction

The HP Embedded Capture (HP EC) Application Programming Interface (API) enables client applications integration that interacts with MFP devices to manage workflow and remote document capture. API services are provided as part of the professional services agreement for HP Embedded Capture 1.1 or higher versions.

1.1 Basic and Advanced modes

The Embedded Capture solution works with a set of FutureSmart and non-FutureSmart MFP devices. FutureSmart offers an advanced set of functionalities that can take advantage of all the power of Embedded Capture. Non-FutureSmart devices cover a subset of those functionalities (basic) with standard document capture capabilities like Scan, metadata (with restrictions), and certain navigation levels (2) that cover the majority of the use cases.

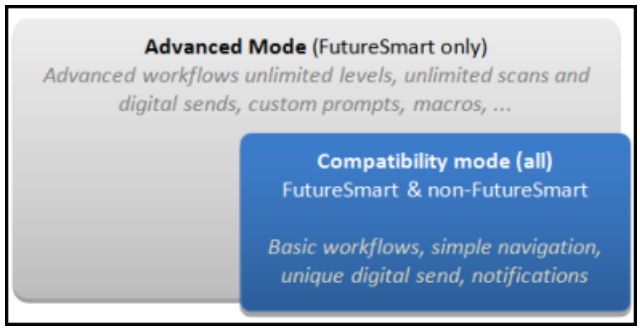
The API is divided into two categories: Basic and Advanced.

1.1.1 Basic API

The Basic API is compatible with the entire fleet (FutureSmart and non-FutureSmart devices). API integrators do not need to distinguish between device models. The same API calls and URLs are available on both models.

1.1.2 Advanced API

The Advanced API extends the complexity and flexibility of workflows managed on a device, offering extra functionalities in addition to what the Basic API provides.



1.2 Accessing the API on a device (%API_URL%)

The Embedded Capture API is exposed throughout the MFP in specific URLs by using SSL (recommended).

```
https://mfp_ip_address/hp/device/hp.extensibility.ec.clientservices.api?parameters
```

Although it is possible to use the API in http mode without encryption, this is NOT recommended. Deployment of workflows with associated parameters (including passwords) will be transferred in plain text over the network, and may be exposed to unauthorized access.

```
http://mfp_ip_address/hp/device/hp.extensibility.ec.clientservices.api?parameters
```

Some API calls will send parameters by GET and others by POST. This is specified on each API definition table.

NOTE: Changes to the transport protocol — to use or stop using SSL — should be done during device configuration (Embedded webserver). See the *HP Embedded Capture Admin Guide* for more information.

NOTE: For all API methods described in this document, URLs have been simplified by replacing the value with `%API_URL%`

1.2.1 XML encoding

All API parameters are based on standard XML documents. The conventions used for this XML are the following:

- PascalCase for elements
- camelCase for attributes

Example:

```
<DestinationElement typeAttribute="email" metadataAttribute="true">
```

1.2.2 XSD validation

Embedded Capture API is validated by an XSD schema that is available for downloading from the devices. Each API includes a "schema" section that helps obtain the XSD document in real time.

To identify any issues with content on a client PC, it is highly recommended that XSD schemas be used to validate the content before sending it to the API.

Schema	Request	%API_URL%?api=jobs&schema=put
	Response	%API_URL%?api=jobs&rschema=put

1.2.3 Response codes

Each API call, except the "logging" API's "get" method, response message will include a code and a descriptive message. The message description may change on future releases of HP Embedded Capture (HP EC).

TIP: Any client application using the response information may use the error codes in lace of strings to ensure future compatibility.

```
<?xml version="1.0" encoding="UTF-8"?>
<Response version="1.1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="%API_URL%?api=common&amp;rschema=fault">
  <OperationStatus>
     <Code>-3</Code>
     <Message>Job id does not correspond to a valid job</Message>
  </OperationStatus>
  <Content/>
</Response>
```

2 API security

To avoid unauthorized access, all API calls can be password protected by the administrator. Protecting the API guarantees that the MFP cannot be accessed by any client PC or application that does not know the credentials to execute the API calls.

NOTE: It is highly recommended that the API be protected by setting the access control password.

2.1 Administrator and API accounts

2.1.1 Admin account

The admin account corresponds with the device administrator account credentials (admin). An MFP device needs to be protected with an administrator password so that advanced options, network settings, etc... (embedded webserver) can be accessed.

The administrator completes the following operations during the installation and normal setup of Embedded Capture:

- setButton
- removeButton
- setApiPassword
- put (silent mode)
- resetSolution

It is, however, recommended that a different password be used for standard API calls. This is explained in the following sections.

2.1.1.1 “apiuser” user account

Setting up this account is optional, but highly recommended. Though once set, it is required for all API operations except the ones specified above (setButton, removeButton, setApiPassword, put (silent mode), resetSolution).

2.1.2 Using the API without authentication

If authentication is not used on the API, certain operations still require setup using an administrator password. This is due to the following standard device usage constraints:

- *setButton, removeButton, resetSolution*:

  Administrator user/password is required for execution of these calls. When the solution is installed from the administrator console, a default button is created, and this password is already used in a transparent way for the administrator (specified on the device list).

- *putSilent*:

  Administrator user/password is required to execute this call. Once the solution is installed, the administrator password is remembered by HP Embedded Capture. This is to avoid having to specify a password on the API "put (silent mode)" calls. If the device administrator password is changed, the Embedded Capture “cached” password must be refreshed using ONE of the following two options:

- Execute any API call with basic authentication using admin user/password.

  or

- From the administrator console, edit the admin password in all changed devices, and press the **Remove workflow** button.

2.2 Basic access authentication

Embedded Capture uses *Basic access authentication* for any API operation requiring authentication. To transmit credentials through HTTP, this authentication must be used.

For detailed information about basic access authentication, refer to the following: http://en.wikipedia.org/wiki/Basic_access_authentication

To log in with API user credentials, the user name must be “apiuser” and the password must be the one specified using the **SetApiPassword** operation.

3 Compatible API

3.1 Graph and job services

3.1.1 Put job

Table 3-1 Put a new job

%API_URL%?api=jobs&method=put			
POST	Description	Uploads new scan job (simple workflow) to target MFP. Uploading scan job in compatible mode appends job to existing workflows on device. Uploading scan job with same filtering parameters as an existing one results in two jobs with same menu options displayed on control panel.	
	Authentication	"admin" or "apiuser" basic authentication required for Silent jobs "put" operation. Optional in others.	
	Payload	IN	XML: Job details. Contains: Navigation settings (optional (*)) Scan settings Custom options Metadata options Notification Destination (*) Silent job: If navigation settings element is not defined, job is considered silent. Silent job is scan&send workflow that executes immediately after put operation finishes; no user interaction is possible on device control panel. This is an example of typical use of TWAIN driver.
		OUT	If 200, XML response with job identifier. If 400, XML response with error code.
Response Code		200 OK — success	
		400 Bad request Error code -1: Product not licensed Error code -3: Error parsing XML payload Error code -10: Device is busy (silent mode) Error code -11: Media size unsupported Error code -12: Unexpected error	
		401 Unauthorized access — if basic authentication fails	
		411 Length required — if content length is not or is badly specified	
		500 Internal Server Error — if too many requests are active. Retry recommended.	
Schema	Request	%API_URL%?api=jobs&schema=put	
	Response	%API_URL%?api=jobs&rschema=put	

Figure 3-1 Put job, Request payload example

```
<?xml version="1.0" encoding="UTF-8"?>
<Request version="1.1.0">
   <Job>
      <NavigationSettings>
         <LabelA value="AAA"/>
         <LabelB value="FULL EMAIL JOB"/>
         <JobDescription value="THIS IS MY JOB"/>
         <Permanent value="true"/>
         <HideDeleteButton value="true"/>
         <HideFileName value="true"/>
         <OneButton value="false"/>
      </NavigationSettings>
      <ScanSettings>
         <Type value="tiff"/>
         <Color value="bw"/>
         <Resolution value="150"/>
         <Duplex value="true"/>
         <Source value="auto"/>
         <MediaSize value="a4"/>
         <PageContent value="text"/>
         <QualityMode value="medium"/>
         <Sharpness value="3"/>
         <Darkness value="4"/>
         <BackgroundRemoval value="4"/>
         <Orientation value="landscape"/>
         <Multipage value="true"/>
         <NumOfPages value="3"/>
      </ScanSettings>
      <CustomOptions>
         <Param key="removeBlankPages" value="on"/>
      </CustomOptions>
      <MetadataOptions>
         <Param key="metatadataKey1" value="metadataDefaultValue1"/>
         <Param key="metatadataKey2" value="metadataDefaultValue2"/>
      </MetadataOptions>
      <Notification>
         <Condition>always</Condition>
         <Email>
            <DestAddress value="You@you.com"/>
            <FromAddress value="Me@me.com"/>
            <Port value="7"/>
         </Email>
      </Notification>
      <Destination>
         <Metadata>true</Metadata>
         <Email>
            <FileName value="scanXX"/>
            <DestAddress value="you@you.com"/>
            <FromAddress value="me@me.com"/>
            <CcAddress value="cc@cc.com"/>
            <BccAddress value="bcc@bcc.com"/>
            <Subject value="Here is the scanned image"/>
         </Email>
      </Destination>
   </Job>
</Request>
```

Destination examples

A scan job can be assigned any of the following destinations:

- **Local**

  The Local destination saves scanned documents to the device hard drive. They are not sent out of the device, and can only be recovered through an API *get* operation.

  ```
  <Local/>
  ```

- **Email**

```
<Email>
    <FileName value="scanXX"/>
    <DestAddress value="you@you.com"/>
    <FromAddress value="me@me.com"/>
    <CcAddress value="cc@cc.com"/>
    <BccAddress value="bcc@bcc.com"/>
    <Subject value="Here is the scanned image"/>
    <Port value="25"/>
    <Notification value="true"/>
</Email>
```

- **FTP**

```
<Ftp>
    <Address value="1.2.3.4"/>
    <Port value="21"/>
    <Path value="\\path\path1"/>
    <MetadataPath value="\\path\path2"/>
    <FileName value="scan"/>
    <UserName value="username"/>
    <Password value="pass"/>
</Ftp>
```

- **Network folder**

```
<NetworkFolder>
    <Domain value="WORKGROUP"/>
    <Path value="\\net\path"/>
    <MetadataPath value="\\net\path2"/>
    <FileName value="scan"/>
    <UserName value="username"/>
    <Password value="password"/>
</NetworkFolder>
```

- **Success response example:**

```
<?xml version="1.0" encoding="UTF-8"?>
<Response version="1.1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="%API_URL%?api=jobs&amp;rschema=put">
    <OperationStatus>
        <Code>0</Code>
        <Message>Request Completed</Message>
    </OperationStatus>
    <Content>
        <JobId>35</JobId>
    </Content>
</Response>
```

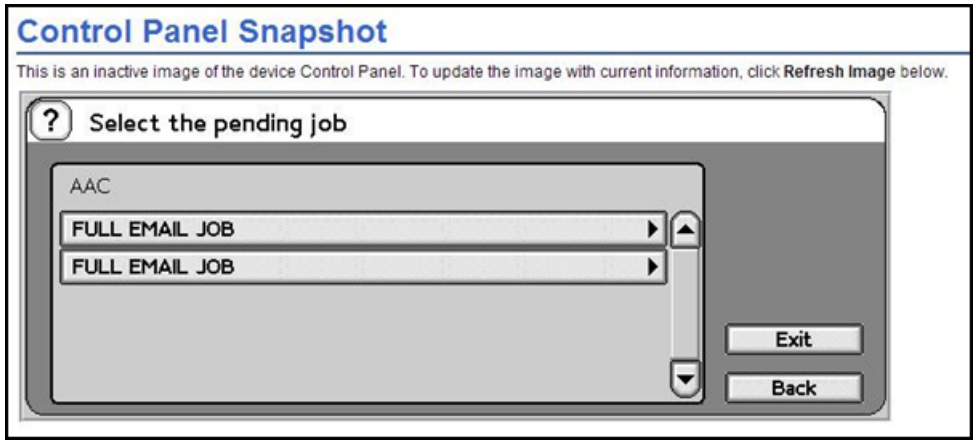
- **Error response example:**

```
<?xml version="1.0" encoding="UTF-8"?>
<Response version="1.1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="%API_URL%?api=jobs&amp;rschema=put">
    <OperationStatus>
        <Code>-3</Code>
        <Message>The given job is invalid</Message>
    </OperationStatus>
    <Content/>
</Response>
```

As an example, when two jobs with the same navigation filters “label A” and “label B” are uploaded, both will be visible under “label A.”

Label A: AAC

Label B: FULL EMAIL JOB



NOTE: See APPENDIX I for possible settings and default values.

3.1.2 View job

Table 3-2 View job

%API_URL%?api=jobs&method=view&jobId={jobId}		
GET	Description	Retrieves the job details as they were set up on the put API call. If the ID is set to 0, returns the details of all the workflow jobs. If the ID is non-zero, returns the details of the job corresponding to the specified id.

Table 3-2 View job (continued)

	Payload	IN	-
		OUT	If 200, XML response with job details. If 400, XML response with error code.
	Response Code	200 OK — success	
		400 Bad request: Error code –2: Invalid job id. Error code –5: Id does not correspond to a job.	
		401 Unauthorized access — if basic authentication fails.	
		500 Internal Server Error — if too many requests are active. Retry recommended.	
Schema	Request	-	
	Response	*API_URL*?api=jobs&rschema=view	

Figure 3-2 View job, success response example

```
<?xml version="1.0" encoding="UTF-8"?>
<Response version="1.1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="&API_URL&?api=jobs&amp;rschema=view">
  <OperationStatus>
    <Code>0</Code>
    <Message>Request Completed</Message>
  </OperationStatus>
  <Content>
    <Job status="pending" id="5" creationDate="04-03-2013 13:15:28">
      <NavigationSettings>
        <LabelA value="AAA"/>
        <LabelB value="FULL EMAIL JOB"/>
        <JobDescription value="THIS IS MY JOB"/>
        <Permanent value="true"/>
        <HideDeleteButton value="true"/>
        <HideFileName value="true"/>
        <OneButton value="false"/>
        <ShowSummaryScreen value="false"/>
      </NavigationSettings>
      <ScanSettings>
        <Type value="tiff"/>
        <Color value="bw"/>
        <Resolution value="150"/>
        <Duplex value="true"/>
        <Source value="auto"/>
        <MediaSize value="a4"/>
        <PageContent value="text"/>
        <QualityMode value="medium"/>
        <Sharpness value="3"/>
        <Darkness value="4"/>
        <BackgroundRemoval value="4"/>
        <Orientation value="landscape"/>
        <Multipage value="true"/>
        <NumOfPages value="3"/>
      </ScanSettings>
      <MetadataOptions>
        <Param key="CUSTOM1" value="MYCUSTOM1"/>
        <Param key="CUSTOM2" value="MYCUSTOM2"/>
      </MetadataOptions>
      <Notification>
        <Condition>always</Condition>
        <Email>
          <DestAddress value="You@you.com"/>
          <FromAddress value="Me@me.com"/>
          <Subject value="NOTIFICATION"/>
        </Email>
      </Notification>
      <Destination>
        <Metadata>true</Metadata>
        <Email>
          <FileName value="scanXX"/>
          <DestAddress value="you@you.com"/>
          <FromAddress value="me@me.com"/>
          <CcAddress value="cc@cc.com"/>
          <BccAddress value="bcc@bcc.com"/>
          <Subject value="Here is the scanned image"/>
        </Email>
      </Destination>
    </Job>
  </Content>
</Response>
```

3.1.3 Delete job

Table 3-3 **Delete job**

&API_URL&?api=jobs&method=delete&jobId={jobId}

Table 3-3 Delete job (continued)

GET	Description	Removes a job by changing its status to “cancelled.” Deleted jobs will still appear in an API view request, but with a cancelled status until they get purged by the Embedded Capture garbage collector. If the jobId is set to 0, all jobs in the graph are removed. If jobId is non-zero, only the specified job is removed.	
	Payload	IN	—
		OUT	If 200, XML response with code 0. If 400, XML response with error code.
	Response Code	200 OK – success	
		400 Bad request: Error code -2: Invalid job id. Error code -5: Id does not correspond to a job. Error code -12: Operation error.	
		401 Unauthorized access – if basic authentication fails.	
		500 Internal Server Error – if too many requests are active. Retry recommended.	
Schema	Request	—	
	Response	%API_URL%?api=common=default	

Figure 3-3 Delete job, success response example

```
<?xml version="1.0" encoding="UTF-8"?>
<Response version="1.1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="%API_URL%?api=common&amp;rschema=default">
 <OperationStatus>
    <Code>0</Code>
    <Message>Request Completed</Message>
 </OperationStatus>
 <Content/>
</Response>
```

3.1.4 Get files

Table 3-4 Get files

%API_URL%?api=jobs&method=getFiles&jobId={jobId}&format}={format}		
GET	Description	Retrieves the files scanned by the job corresponding to the specified id. If the format is set to “zip”, returns all files in a zip. If the format is set to “boundary”, returns all files as a MIME multipart message (http://en.wikipedia.org/wiki/MIME). If the format is set to “links”, returns an xml with direct links to download files. For silent jobs (no user interaction on the control panel) the getFiles is a blocking operation, i.e. no result is returned until files are ready or the job is cancelled.

Table 3-4 Get files (continued)

		Metadata files not fetched when getting files from an API Call after putting a local job with metadata true.	
	Payload	IN	—
		OUT	If 200, XML response with files or xml with direct download links. If 400, XML response with error code.
	Response Code	200 OK – success	
		400 Bad request: Error code -2: Invalid job id Error code -5: Id does not correspond to a job Error code -6: Error loading job information Error code -7: Error creating zip	
		401 Unauthorized access – if basic authentication fails	
		500 Internal Server Error – if too many requests are active. Retry recommended.	
Schema	Request	—	
	Response	%API_URL%?api=jobs&rschema=getFiles	

Figure 3-4 Get files, response example (Applicable only for format=links)

```
<?xml version="1.0" encoding="UTF-8"?>
<Response version="1.1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="%API_URL%?api=jobs&amp;rschema=getFiles">
  <OperationStatus>
    <Code>0</Code>
    <Message>Request Completed</Message>
  </OperationStatus>
  <Content>
    <Document>%API_URL%?method=getFiles&amp;jobId={jobId}&amp;page=1</Document>
    <Document>%API_URL%?method=getFiles&amp;jobId={jobId}&amp;page=2</Document>
    <Document>%API_URL%?method=getFiles&amp;jobId={jobId}&amp;page=3</Document>
    <Document>%API_URL%?method=getFiles&amp;jobId={jobId}&amp;page=4</Document>
    <Document>%API_URL%?method=getFiles&amp;jobId={jobId}&amp;page=5</Document>
    <Document>%API_URL%?method=getFiles&amp;jobId={jobId}&amp;page=6</Document>
  </Content>
</Response>
```

3.1.5 Set purge settings

Table 3-5 Set purge settings

%API_URL%?api=jobs&method=setPurgeSettings		
POST	Description	Sets the Expiration time and the Garbage Collector period for dynamic scan jobs in the MFP. These variables control the time that a job will remain in the device. Expiration time corresponds to the amount of time that dynamic jobs remain in the device. Collector period corresponds to time interval within subsequent cleanup operations of expired jobs. Collector period must be smaller than Expiration time. Both values are defined in seconds.

Table 3-5 Set purge settings (continued)

		By default, Collector period is set to 30 Minutes and Expiration Time to 12 Hours. Collector period valid Range is between 1800 sec (30 minutes) and 86400 sec (24 hours). Expiration Time is not constrained.	
	Payload	IN	XML Configuration parameters
		OUT	If 200, XML response with code 0 If 400, XML response with error code
	Response Code	200 OK – success	
		400 Bad request: Error code -3: Error parsing xml payload	
		401 Unauthorized access – if basic authentication fails	
		411 Length required – if content length is not or is badly specified	
		500 Internal Server Error – if too many requests are active. Retry recommended.	
Schema	Request	%API_URL%?api=jobs&schema=setPurgeSettings	
	Response	%API_URL%?api=common&rschema=default	

Figure 3-5 Set purge settings, request payload example

```
<?xml version="1.0" encoding="UTF-8"?>
<Request version="1.1.0">
    <ExpirationTime>10000</ExpirationTime>
    <CollectorPeriod>3600</CollectorPeriod>
</Request>
```

Figure 3-6 Set purge settings, success response example

```
<?xml version="1.0" encoding="UTF-8"?>
<Response version="1.1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="%API_URL%?api=common&amp;rschema=default">
   <OperationStatus>
      <Code>0</Code>
      <Message>Request Completed</Message>
   </OperationStatus>
   <Content/>
</Response>
```

3.2 Configuration services

3.2.1 Get device info

Table 3-6 Get device info

%API_URL%?api=config&method=getDeviceInfo		
GET	Description	Gets the device information. Information returned includes:

Table 3-6 Get device info (continued)

		◦ Device model	
		◦ Device IP	
		◦ Device Family (“FutureSmart” or “Non-FutureSmart”)	
		◦ Hostname	
		◦ Tray width (mm)	
		◦ Tray height (mm)	
	Payload	IN	—
		OUT	If 200, XML response with device info.
	Response Code	200 OK — success	
		401 Unauthorized access — if basic authentication fails	
		500 Internal Server Error — if too many requests are active. Retry recommended.	
Schema	Request	—	
	Response	%API_URL%?api=config&rschema=getDeviceInfo	

Figure 3-7 Get device info, success response example

```
<?xml version="1.0" encoding="UTF-8"?>
<Response version="1.1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="%API_URL%?api=config&amp;rschema=getDeviceInfo">
   <OperationStatus>
      <Code>0</Code>
      <Message>Request Completed</Message>
   </OperationStatus>
   <Content>
      <DeviceInfo>
         <Model>CM3530</Model>
         <Family>Non-Futuresmart</Family>
         <IP>11.11.11.11</IP>
         <Hostname>device.hp.corp.net</Hostname>
         <Tray>
            <Width>216</Width>
            <Height>400</Height>
         </Tray>
      </DeviceInfo>
   </Content>
</Response>
```

3.2.2 Get device status

Table 3-7 Get device status

%API_URL%?api=config&method=getDeviceStatus		
GET	Description	Returns device status information. The information returned includes: ◦ Disk space in bytes. ◦ ADF status

Table 3-7 Get device status (continued)

		◦ Flatbed status	
	Payload	IN	—
		OUT	If 200, XML response with device status.
	Response Code	200 OK — success	
		401 Unauthorized access — if basic authentication fails	
		500 Internal Server Error — if too many requests are active. Retry recommended.	
Schema	Request	—	
	Response	%API_URL%?api=config&rschema=getDeviceStatus	

Figure 3-8 Get device status, response example

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<Response version="1.2.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="%API_URL%?api=config&amp;rschema=getdevicestatus">
  <OperationStatus>
    <Code>0</Code>
    <Message>Request Completed.</Message>
  </OperationStatus>
  <Content>
    <DeviceStatus>
      <DiskAvailable>10024681472</DiskAvailable>
      <AdfStatus>0</AdfStatus>
      <FlatbedStatus>-2</FlatbedStatus>
    </DeviceStatus>
  </Content>
</Response>
```

Table 3-8 ADF Status possible values

Code	Meaning	Explanation
1	Ready	There is paper for scan
0	Empty	There is no paper for scan
-1	Initializing	Device is still booting
–2	Unsupported	Device does not support ADF/Flatbed status monitoring
-3	Not present	Device has no ADF/Flatbed

3.2.3 Get solution info

Table 3-9 Get solution info

%API_URL%?api=config&method=getSolutionInfo		
GET	Description	Returns solution information. Information returned includes: Solution version License information

Table 3-9 Get solution info (continued)

		Blocked for users (through the accessibility block command)	
		Purge settings	
		Log level (all, off)	
		AdvancedWorkflowSupport (true, false)	
	Payload	IN	—
		OUT	If 200, XML response with solution info.
	Response Code	200 OK — success	
		401 Unauthorized access — if basic authentication fails	
		500 Internal Server Error — if too many request are active. Retry recommended.	
Schema	Request	—	
	Response	%API_URL%?api=config&rschema=getSolutionInfo	

Figure 3-9 Get solution info, response example

```
<?xml version="1.0" encoding="UTF-8"?>
<Response xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="%API_URL%?api=config&rschema=getsolutioninfo" version="1.5.0">
    <OperationStatus>
        <Code>0</Code>
        <Message>Request completed.</Message>
    </OperationStatus>
    <Content>
        <SolutionInfo>
            <Version>1.5.0</Version>
            <IsLicensed>false</IsLicensed>
            <IsBlocked>false</IsBlocked>
            <LogLevel>all</LogLevel>
            <AdvancedWorkflowSupport>true</AdvancedWorkflowSupport>
-           <PurgeSettings>
                <ExpirationTime>43200</ExpirationTime>
                <CollectorPeriod>1800</CollectorPeriod>
            </PurgeSettings>
        </SolutionInfo>
    </Content>
</Response>
```

3.2.4 Get solution status

Table 3-10 Get solution status

%API_URL%?api=config&method=getSolutionStatus			
GET	Description	Returns device status information. Information returned includes:	
		Operating status (Navigating, scanning, processing, unknown)	
		Error condition indicating whether the status is at the moment interrupted by some error condition. For example, if there is no paper on ADF.	
	Payload	IN	—
		OUT	If 200, XML response with solution status.

Table 3-10 Get solution status (continued)

	Response Code	200 OK — success
		401 Unauthorized access — if basic authentication fails.
		500 Internal Server Error — if too many requests are active. Retry recommended.
Schema	Request	—
	Response	%API_URL%?api=config&rschema=getSolutionStatus

Figure 3-10 Get solution status, Response example

```
<?xml version="1.0" encoding="UTF-8"?>
<Response version="1.1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="%API_URL%?api=config&amp;rschema=getSolutionStatus">
   <OperationStatus>
      <Code>0</Code>
      <Message>Request Completed</Message>
   </OperationStatus>
   <Content>
      <SolutionStatus>
         <OperatingStatus>2</OperatingStatus>
         <ErrorCondition>false</ErrorCondition>
      </SolutionStatus>
   </Content>
</Response>
```

Table 3-11 Navigation status possible values

Code	Meaning
0	Unknown
1	Navigating
2	Scanning
3	Processing
4	Idle

3.2.5 Wake up

Table 3-12 Wake up

%API_URL%?api=config&method=wakeup			
GET	Description	Wakes up device if in standby mode. If device is in standby mode, starting the operation with Embedded Capture is delayed by the wake up process. With this API call it is possible to force the wake up programmatically, so that it is ready when the user arrives.	
	Payload	IN	—
		OUT	If 200, XML response with current status.
	Response Code	200 OK — success	
		401 Unauthorized access — if basic authentication fails	

Table 3-12 Wake up (continued)

		500 Internal Server Error — if too many requests are active. Retry recommended.
Schema	Request	—
	Response	%API_URL%?api=config&rschema=wakeup

Figure 3-11 Wake up, success response example

```
<?xml version="1.0" encoding="UTF-8"?>
<Response version="1.1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="%API_URL%?api=config&amp;rschema=wakeup">
   <OperationStatus>
      <Code>0</Code>
      <Message>Request Completed</Message>
   </OperationStatus>
   <Content>
      <DeviceStatus>1</DeviceStatus>
   </Content>
</Response>
```

Table 3-13 Device Status possible values

Code	Meaning	Explanation
1	OK	Device is awake
0	KO	Device is not awake
-1	Initializing	Device is still booting

3.2.6 Cancel scan

Table 3-14 Cancel scan

%API_URL%?api=config&method=cancelScan			
GET	**Description**	Cancels/interrupts the scanning process on the device.	
	Payload	IN	—
		OUT	If 200, XML response with code 0 If 400, XML response with error code
	Response Code	200 OK — success	
		400 Bad request: Error code –10: Device is busy	
		401 Unauthorized access — if basic authentication fails	
		500 Internal Server Error — if too many request are active. Retry recommended.	
Schema	Request	—	
	Response	%API_URL%?api=common&rschema=default	

Figure 3-12 Cancel scan, Success response example

```
<?xml version="1.0" encoding="UTF-8"?>
<Response version="1.1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="%API_URL%?api=common&amp;rschema=default">
   <OperationStatus>
      <Code>0</Code>
      <Message>Request Completed</Message>
   </OperationStatus>
   <Content/>
</Response>
```

3.2.7 Reset Solution

Table 3-15 Reset Solution

%API_URL%?api=config&method=resetSolution			
GET	Description	This function restores the solution as if it was newly installed on a clean device: ◦ Removes all process data such as scanned files, pending jobs etc. ◦ Restores default solution settings: Deactivates logs, removes icon, resets purge settings and API passwords. **WARNING!** Removing the solution button will cause all access control configuration to be lost. Upon creating a new button, access will have to be reconfigured.	
	Authentication	Requires basic authentication with device admin credentials.	
	Payload	IN	—
		OUT	If 200, XML response with code 0 If 400, XML response with error code
	Response Code	200 OK — success	
		400 Bad request: Error code –10: Device is busy Error code –12: Unexpected error	
		401 Unauthorized access – if basic authentication fails	
		500 Internal Server Error – if too many requests are active. Retry recommended.	
Schema	Request	—	
	Response	%API_URL%?api=common&rschema=default	

Figure 3-13 Reset solution, success response example

```
<?xml version="1.0" encoding="UTF-8"?>
<Response version="1.1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="%API_URL%?api=common&amp;rschema=default">
   <OperationStatus>
      <Code>0</Code>
      <Message>Request Completed</Message>
   </OperationStatus>
   <Content/>
</Response>
```

3.3 Extensibility services

3.3.1 Set button

Table 3-16 Set button

%API_URL%?api=extensibility&method=setButton			
POST	Description	Creates or modifies the Home screen button on the device. If any fields are not specified, a default value will be used. If Embedded Capture had no button (silent mode), it will change to a non-silent mode (interactive) after the button creation. **Icon requirements:** Family / Type / Size (width x heigh) NFsmart10 / GIF / 106x149px NFsmart40 / GIF / 60x55px NFsmart / GIF / 48x44px FSMART / GIF / 66x66px	
	Authentication	Requires basic authentication using device admin credentials.	
	Payload	IN	XML button details
		OUT	If 200, XML response with code 0 If 400, response with error code
	Response Code	200 OK – success	
		400 Bad request: Error code -3: Error parsing xml payload Error code -10: Device is busy Error code -12: Unexpected error	
		401 Unauthorized access – if basic authentication fails	
		411 Length required – if content length is not or is badly specified	
		500 Internal server error – if too many requests are active. Retry recommended.	
Schema	Request	%API_URL%?api=extensibility&schema=setButton	
	Response	%API_URL%?api=common&rschema=default	

Figure 3-14 Extensibility services, Set button — request payload example

```
<?xml version="1.0" encoding="UTF-8"?>
<Request version="1.1.0">
    <Button>
        <Title>EC</Title>
        <Description>Title</Description>
        <Icons>
            <NFsmart>ImageEncodedInBase64</NFsmart>
            <NFsmart10>ImageEncodedInBase64</NFsmart10>
            <NFsmart40>ImageEncodedInBase64</NFsmart40>
            <Fsmart>ImageEncodedInBase64</Fsmart>
        </Icons>
    </Button>
</Request>
```

Figure 3-15 Extensibility services, Set button — success response example

```
<?xml version="1.0" encoding="UTF-8"?>
<Response version="1.1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="%API_URL%?api=common&amp;rschema=default">
   <OperationStatus>
      <Code>0</Code>
      <Message>Request Completed</Message>
   </OperationStatus>
   <Content/>
</Response>
```

3.3.2 Remove button

Table 3-17 Remove button

%API_URL%?api=extensibility&method=removeButton			
GET	Description	Removes a Home screen button on the device. Attention: Removing the solution button will cause all access control configuration to be lost (authentication agent, embedded authentication...). Upon creating a new button, the application access will need to be reconfigured.	
	Authentication	Requires basic authentication with device admin credentials.	
	Payload	IN	—
		OUT	If 200, XML response with code 0 If 400, XML response with error code
	Response Code	200 OK — success	
		400 Bad request Error code -10: Device is busy Error code -12: Unexpected error	
		401 Unauthorized access – if basic authentication fails	
		500 Internal Server Error – if too many requests are active. Retry recommended.	

Table 3-17 Remove button (continued)

Schema	Request	—
	Response	%API_URL%?api=common&rschema=default

Figure 3-16 Remove button, success response example

```
<?xml version="1.0" encoding="UTF-8"?>
<Response version="1.1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="%API_URL%?api=common&amp;rschema=default">
   <OperationStatus>
      <Code>0</Code>
      <Message>Request Completed</Message>
   </OperationStatus>
   <Content/>
</Response>
```

3.4 Accessibility services

3.4.1 Set API password

Table 3-18 Set API Password

%API_URL%?api=accessibility&method=setApiPassword

POST	Description	Sets a password for API authentication. By default, password is not set, and therefore API is not protected. For security reasons, it is recommended that API protection be used to avoid unauthorized access to scanned documents or workflows information. Once API password is set, all operations will require basic authentication with credentials: **Username:** “apiuser” **Password:** the password defined. (Alternatively, device "admin" user/password can be used for API authentication.) To unset API password, an empty string must be unspecified in payload xml.	
	Authentication	Requires basic authentication with device admin credentials.	
	Payload	IN	XML: Containing the password of the administrator of the API
		OUT	If 200, XML response with code 0 If 400, XML response with error code
	Response Code	200 OK – success	
		400 Bad request Error code -3: Error parsing xml payload	
		401 Unauthorized access – if basic authentication fails	
		411 Length required – if content length is not or is badly specified	
		500 Internal Server Error – if too many requests are active. Retry recommended.	

Table 3-18 Set API Password (continued)

Schema	Request	%API_URL%?api=accessibility&schema=setApiPassword
	Response	%API_URL%?api=common&rschema=default

Figure 3-17 Set API password, request payload example

```
<?xml version="1.0" encoding="UTF-8"?>
<Request version="1.1.0">
    <ApiPassword>API Password</ApiPassword>
</Request>
```

Figure 3-18 Set API password, success response example

```
<?xml version="1.0" encoding="UTF-8"?>
<Response version="1.1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="%API_URL%?api=common&amp;rschema=default">
   <OperationStatus>
      <Code>0</Code>
      <Message>Request Completed</Message>
   </OperationStatus>
   <Content/>
</Response>
```

NOTE: When the API password is set, API calls from EC Installer do not work. This is because EC Installer doesn't use/recognize the API password.

3.4.2 Block Embedded Capture UI

Table 3-19 Block device

%API_URL%?api=accessibility&method=block			
GET	Description	Disables the Embedded Capture Home screen button on the control panel. This is highly recommended before performing administration tasks that need to change workflows structure and may affect the user scanning.	
	Payload	IN	—
		OUT	If 200, XML response with code 0. If 400, XML response with error code.
	Response Code	200 OK – success	
		400 Bad request: Error code –10: Device is busy. Error code –12: Unexpected error.	
		401 Unauthorized access – if basic authentication fails	
		500 Internal Server Error – if too many requests are active. Retry recommended.	
Schema	Request	—	
	Response	%API_URL%?api=common&rschema=default	

Figure 3-19 Block Embedded Capture UI, success response example

```
<?xml version="1.0" encoding="UTF-8"?>
<Response version="1.1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="%API_URL%?api=common&amp;rschema=default">
   <OperationStatus>
      <Code>0</Code>
      <Message>Request Completed</Message>
   </OperationStatus>
   <Content/>
</Response>
```

3.4.3 Unblock Embedded Capture UI

Table 3-20 Unblock device

%API_URL%?api=accessibility&method=unblock			
GET	Description	Reactivates solution button on device control panel. After unblocking Embedded Capture UI, users will be able to execute workflows normally.	
	Payload	IN	—
		OUT	If 200, XML response with code 0 If 400, XML response with error code
	Response Code	200 OK – success	
		400 Bad request: Error code -10: Device is busy Error code -12: Unexpected error	
		401 Unauthorized access – if basic authentication fails	
		500 Internal Server Error – if too many requests are active. Retry recommended.	
Schema	Request	—	
	Response	%API_URL%?api=common&rschema=default	

Figure 3-20 Unblock Embedded Capture UI, success response example

```
<?xml version="1.0" encoding="UTF-8"?>
<Response version="1.1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="%API_URL%?api=common&amp;rschema=default">
   <OperationStatus>
      <Code>0</Code>
      <Message>Request Completed</Message>
   </OperationStatus>
   <Content/>
</Response>
```

3.5 Logging services

3.5.1 Enable log

Table 3-21 Enable log

&API_URL&?api=logging&method=enable			
POST	Description	Method that enables the Logging Service during a specified number of days. If specifying 0 Days, the Logging Service will be enabled permanently. **NOTE:** Take into consideration that enabling the log permanently will shorten the printer's hard disk lifetime, and may also affect performance.	
	Payload	IN	XML Configuration parameters
		OUT	If 200, XML response with code 0 If 400, XML response with error code
	Response code	200 OK – success	
		400 Bad request: Error code -3: Error parsing xml payload	
		401 Unauthorized access – if basic authentication fails	
		411 Length required – if content length is not or is badly specified.	
		500 Internal Server Error – if too many requests are active. Retry recommended.	
Schema	Request	&API_URL&?api=logging&schema=enable	
	Response	&API_URL&?api=common&rschema=default	

Figure 3-21 Enable log, request payload example

```
<?xml version="1.0" encoding="UTF-8"?>
<Request version="1.1.0">
  <NumberOfDays>2</NumberOfDays>
</Request>
```

Figure 3-22 Enable log, success response example

```
<?xml version="1.0" encoding="UTF-8"?>
<Response version="1.1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="&API_URL&?api=common&amp;rschema=default">
  <OperationStatus>
     <Code>0</Code>
     <Message>Request Completed</Message>
  </OperationStatus>
  <Content/>
</Response>
```

3.5.2 Get log

Table 3-22 Get log

%API_URL%?api=logging&method=get			
GET	Description	Method that retrieves the logs of Embedded Capture. The logs are returned in a text file that is downloaded by http protocol.	
	Payload	IN	—
		OUT	If 200, response with the file
	Response code	200 OK — success	
		401 Unauthorized access – if basic authentication fails	
		500 Internal Server Error – if too many requests are active. Retry recommended.	
Schema	Request	—	
	Response	%API_URL%?api=common&rschema=default	

3.5.3 Disable log

Table 3-23 Disable log

%API_URL%?api=logging&method=disable			
GET	Description	Disable the Logging Service.	
	Payload	IN	—
		OUT	If 200, XML response with code 0
	Response Code	200 OK – success	
		401 Unauthorized access – if basic authentication fails	
		500 Internal Server Error – if too many requests are active. Retry recommended.	
Schema	Request	—	
	Response	%API_URL%?api=common&rschema=default	

Figure 3-23 Disable log, success response example

```
<?xml version="1.0" encoding="UTF-8"?>
<Response version="1.1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="%API_URL%?api=common&amp;rschema=default">
   <OperationStatus>
      <Code>0</Code>
      <Message>Request Completed</Message>
   </OperationStatus>
   <Content/>
</Response>
```

4 Advanced API

The Advanced mode API is only available on FutureSmart devices.

To distinguish between device models when using Advanced API calls in a mixed fleet, it is highly recommended that you use the getDeviceInfo API call (<Family> element) on Compatible mode to filter and choose the devices that will accept the advanced calls between the ones that would reject them.

4.1 Graph and job services

4.1.1 Set graph

Table 4-1 Set graph

%API_URL%?api=graph&method=set			
POST	Description	Creates a graph on target MFP. A graph is represented on the MFP as a workflow that may include all its components. This operation replaces the previous graph on the MFP.	
	Payload	IN	XML: Graph
		OUT	If 200, XML response with code 0 If 400, XML response with error code
	Response Code	200 OK – success	
		400 Bad request Error code -1: Product not licensed Error code -3: Error parsing xml payload Error code -8: Unsupported scan settings Error code -10: Device is busy (silent mode) Error code -12: Unexpected error	
		401 Unauthorized access – if basic authentication fails	
		411 Length required – if content length is not or is badly specified.	
		500 Internal Server Error – if too many requests are active. Retry recommended.	
Schema	Request	%API_URL%?api=graph&schema=set	
	Response	%API_URL%?api=common&rschema=default	

Figure 4-1 Set graph, request payload example

```
<?xml version="1.0" encoding="UTF-8"?>
<Request version="1.1.0">
    <Graph>
        <MenuOption name="001_TypicalGraph" disableInactivityTimeout="False" id="deaca780-f7cf-47e1-
a3f8-822b7f730615">
            <Description>Select an option</Description>
            <Buttons>
                <Button action="Back" label="Back"/>
                <Button action="Exit" label="Exit"/>
            </Buttons>
            <Edges>
                <RootParent/>
                <Parent id="7361fa2d-6c6f-4af6-9519-83f0b167e16c"/>
            </Edges>
        </MenuOption>
        <MenuOption name="ScreenSelection" disableInactivityTimeout="False" id="d327059c-0980-4322-
9272-c3d9b8accdfd">
            <Description>Select an option</Description>
            <Buttons>
                <Button action="Back" label="backMn"/>
                <Button action="Exit" label="exitMn"/>
            </Buttons>
            <Edges>
                <Parent id="deaca780-f7cf-47e1-a3f8-822b7f730615"/>
            </Edges>
        </MenuOption>
        <Form name="Metadata Name" disableInactivityTimeout="False" id="5e1253dc-902c-4f49-a0ed-
7411001e6b64">
            <Description>Metadata Title</Description>
            <Buttons>
                <Button action="Back" label="Back"/>
                <Button action="Next" label="Next"/>
            </Buttons>
            <FormOptions>
                <FormOption key="metadataText" name="Metadata Text" type="Text">
                    <Visibility>ReadWrite</Visibility>
                    <Description>Example of text</Description>
                    <DefaultValue>random text</DefaultValue>
                    <MinValue>5</MinValue>
                    <MaxValue>15</MaxValue>
                    <CustomRegularExpression/>
                    <Style/>
                    <IsPassword>False</IsPassword>
                </FormOption>
                <FormOption key="metadataRegExprIP" name="Custom Expr (IP)" type="Text">
                    <Visibility>ReadWrite</Visibility>
                    <Description>Example of IP Validation</Description>
                    <DefaultValue>15.16.17.18</DefaultValue>
                    <MinValue>-2.147484E+09</MinValue>
                    <MaxValue>2.147484E+09</MaxValue>
                    <CustomRegularExpression>^(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-
4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-
9][0-9]?)$</CustomRegularExpression>
                    <Style/>
                    <IsPassword>False</IsPassword>
                </FormOption>
                <FormOption key="metadataRegExprDNI" name="Custom Expr (DNI)" type="Text">
                    <Visibility>ReadWrite</Visibility>
                    <Description>Example of DNI Validation</Description>
                    <DefaultValue>12345678A</DefaultValue>
                    <MinValue>-2.147484E+09</MinValue>
                    <MaxValue>2.147484E+09</MaxValue>
                    <CustomRegularExpression>^[0-9]{8}[a-zA-Z]$</CustomRegularExpression>
                    <Style/>
                    <IsPassword>False</IsPassword>
                </FormOption>
```

```
            <FormOption key="metadataMail" name="Metadata Mail" type="Text">
               <Visibility>ReadWrite</Visibility>
               <Description>Example of mail</Description>
               <DefaultValue>albertv@hp.com</DefaultValue>
               <MinValue>-2.147484E+09</MinValue>
               <MaxValue>2.147484E+09</MaxValue>
               <CustomRegularExpression>^([a-zA-Z0-9_\.\-])+\@(([a-zA-Z0-9\-])+\.)+([a-zA-Z0-
9]{2,4})+$</CustomRegularExpression>
               <Style/>
               <IsPassword>False</IsPassword>
            </FormOption>
            <FormOption key="metadataInt" name="Metadata Int" type="Int">
               <Visibility>ReadWrite</Visibility>
               <Description>Example of int</Description>
               <DefaultValue>5</DefaultValue>
               <MinValue>5</MinValue>
               <MaxValue>15</MaxValue>
               <CustomRegularExpression/>
               <Style/>
               <IsPassword>False</IsPassword>
            </FormOption>
            <FormOption key="metadataDecimal" name="Metadata Decimal" type="Decimal">
               <Visibility>ReadWrite</Visibility>
               <Description>Example of Decimal</Description>
               <DefaultValue>5.25</DefaultValue>
               <MinValue>5</MinValue>
               <MaxValue>15</MaxValue>
               <CustomRegularExpression/>
               <Style/>
               <IsPassword>False</IsPassword>
            </FormOption>
            <FormOption key="metadataBool" name="Metadata Bool" type="Bool">
               <Visibility>ReadWrite</Visibility>
               <Description>Example of Bool</Description>
               <DefaultValue>true</DefaultValue>
               <MinValue>1</MinValue>
               <MaxValue>1</MaxValue>
               <CustomRegularExpression/>
               <Style/>
               <IsPassword>False</IsPassword>
            </FormOption>
            <FormOption key="metadataComboList" name="Metadata Combo List" type="ComboList">
               <Visibility>ReadWrite</Visibility>
               <Description>Example of Combo List</Description>
               <DefaultValue>value 2</DefaultValue>
               <MinValue>5</MinValue>
               <MaxValue>15</MaxValue>
               <CustomRegularExpression/>
               <Style/>
               <IsPassword>False</IsPassword>
               <PossibleValues>
                  <PossibleValue>value 1</PossibleValue>
                  <PossibleValue>value 2</PossibleValue>
                  <PossibleValue>value 3</PossibleValue>
                  <PossibleValue>value 4</PossibleValue>
               </PossibleValues>
            </FormOption>
            <FormOption key="metadataPassword" name="Metadata Password" type="Text">
               <Visibility>ReadWrite</Visibility>
               <Description>Example of password</Description>
               <DefaultValue>random Pass</DefaultValue>
               <MinValue>5</MinValue>
               <MaxValue>15</MaxValue>
               <CustomRegularExpression/>
               <Style/>
               <IsPassword>True</IsPassword>
            </FormOption>
            <FormOption key="metadataRadioList" name="Metadata Radio List" type="RadioList">
               <Visibility>ReadWrite</Visibility>
```

```
                <Description>Example of Radio List</Description>
                <DefaultValue>value 2</DefaultValue>
                <MinValue>5</MinValue>
                <MaxValue>15</MaxValue>
                <CustomRegularExpression/>
                <Style/>
                <IsPassword>False</IsPassword>
                <PossibleValues>
                   <PossibleValue>value 1</PossibleValue>
                   <PossibleValue>value 2</PossibleValue>
                   <PossibleValue>value 3</PossibleValue>
                   <PossibleValue>value 4</PossibleValue>
                </PossibleValues>
             </FormOption>
          </FormOptions>
          <Edges>
             <Parent id="d327059c-0980-4322-9272-c3d9b8acdfd"/>
          </Edges>
       </Form>
       <ScanOptionsScreen name="ScanOptionsScreen" disableInactivityTimeout="False" id="2b94b08b-
7cdd-46c5-90ea-bd3a3969c92b">
          <Description>Select scan options</Description>
          <Buttons>
             <Button action="Back" label="Back"/>
             <Button action="Next" label="Next"/>
          </Buttons>
          <ScanSettings>
             <Resolution label="Resolution">
                <Visibility>ReadWrite</Visibility>
                <DefaultValue>Res150x150</DefaultValue>
             </Resolution>
             <PreviewMode label="Preview Mode">
                <Visibility>ReadOnly</Visibility>
                <DefaultValue>Off</DefaultValue>
             </PreviewMode>
             <MediaOrientation label="Media Orientation">
                <Visibility>ReadWrite</Visibility>
                <DefaultValue>Landscape</DefaultValue>
             </MediaOrientation>
             <CustomLength label="Custom Length">
                <Visibility>ReadOnly</Visibility>
                <DefaultValue>2,9</DefaultValue>
             </CustomLength>
             <ColorMode label="Color Mode">
                <Visibility>ReadWrite</Visibility>
                <DefaultValue>Color</DefaultValue>
             </ColorMode>
             <FileType label="File Type">
                <Visibility>ReadWrite</Visibility>
                <DefaultValue>Jpeg</DefaultValue>
             </FileType>
             <MediaSize label="Media Size">
                <Visibility>ReadOnly</Visibility>
                <DefaultValue>A3</DefaultValue>
             </MediaSize>
             <PdfCompressionMode label="Pdf Compression Mode">
                <Visibility>ReadWrite</Visibility>
                <DefaultValue>Normal</DefaultValue>
             </PdfCompressionMode>
             <TiffCompressionMode label="Tiff Compression Mode">
                <Visibility>ReadWrite</Visibility>
                <DefaultValue>NotApplicable</DefaultValue>
             </TiffCompressionMode>
             <XpsCompressionMode label="XPS Compression Mode">
                <Visibility>ReadWrite</Visibility>
```

```
                <DefaultValue>NotApplicable</DefaultValue>
            </XpsCompressionMode>
            <OcrLanguage label="Ocr Language">
                <Visibility>ReadWrite</Visibility>
                <DefaultValue>NotApplicable</DefaultValue>
            </OcrLanguage>
            <PdfEncryptionPassword label="Pdf Encryption Password">
                <Visibility>ReadWrite</Visibility>
                <DefaultValue>MyPass</DefaultValue>
            </PdfEncryptionPassword>
        </ScanSettings>
        <Edges>
            <Parent id="5e1253dc-902c-4f49-a0ed-7411001e6b64"/>
        </Edges>
    </ScanOptionsScreen>
    <ScanProcess name="Node" id="cafcd914-2468-4333-89ae-094ed79f467b">
        <Settings>
            <ShowProgressScreen>True</ShowProgressScreen>
            <WaitingText>Scanning...</WaitingText>
            <ShowWaitingImage>True</ShowWaitingImage>
        </Settings>
        <BasicOptions>
            <BackgroundCleanup>Value0</BackgroundCleanup>
            <BlankImageRemovalMode>Off</BlankImageRemovalMode>
            <ColorDropoutMode>Off</ColorDropoutMode>
            <ColorMode>Color</ColorMode>
            <ContrastAdjustment>Value0</ContrastAdjustment>
            <CropMode>Off</CropMode>
            <CustomLength>2.9</CustomLength>
            <CustomWidth>2.16</CustomWidth>
            <DarknessAdjustment>Value0</DarknessAdjustment>
            <DuplexFormat>Flip</DuplexFormat>
            <FileType>Jpeg</FileType>
            <JobAssemblyMode>Off</JobAssemblyMode>
            <MediaOrientation>Landscape</MediaOrientation>
            <MediaSize>A4</MediaSize>
            <MediaSource>Flatbed</MediaSource>
            <MediaWeightAdjustment>NotApplicable</MediaWeightAdjustment>
            <MisfeedDetectionMode>Off</MisfeedDetectionMode>
            <OutputQuality>Medium</OutputQuality>
            <PlexMode>Simplex</PlexMode>
            <PreviewMode>Off</PreviewMode>
            <ProgressDialogMode>Off</ProgressDialogMode>
            <Resolution>Res150x150</Resolution>
            <SharpnessAdjustment>Value0</SharpnessAdjustment>
            <TextPhotoOptimization>Mixed1</TextPhotoOptimization>
        </BasicOptions>
        <FileOptions>
            <PdfEncryptionPassword/>
            <OcrLanguage>NotApplicable</OcrLanguage>
            <PdfCompressionMode>Normal</PdfCompressionMode>
            <TiffCompressionMode>NotApplicable</TiffCompressionMode>
            <XpsCompressionMode>Normal</XpsCompressionMode>
        </FileOptions>
        <NSEOptions>
            <NumPages>-1</NumPages>
            <Permanent>True</Permanent>
        </NSEOptions>
        <Edges>
            <Parent id="2b94b08b-7cdd-46c5-90ea-bd3a3969c92b"/>
        </Edges>
    </ScanProcess>
    <DialogScreen name="Node" disableInactivityTimeout="False" id="7361fa2d-6c6f-4af6-9519-
83f0b167e16c">
        <Description>Message</Description>
        <Buttons>
```

```
            <Button action="Back" label="Back"/>
            <Button action="Exit" label="Exit"/>
            <Button action="Next" label="Next"/>
         </Buttons>
         <Message>Document file type: %_Scan.ScanBasicOptions.FileType_%&lt;br/&gt;Scanned pages:
%_Scan.ScanNumPages_%&lt;br/&gt;Scanned files: %_Scan.FileCount_%&lt;br/&gt;&lt;br/&gt;&lt;WHILE
type="process" scope="cycle" delimiter="&lt;br/&gt;"&gt;  %_Process.Name_%:&lt;br/&gt;
%_Process.Result_% - %_Process.ResultMessage_%&lt;br/&gt;     (Files:
%_Process.FileCount_%)&lt;br/&gt;&lt;/WHILE&gt;</Message>
         <Edges>
            <Parent id="2b94b08b-7cdd-46c5-90ea-bd3a3969c92b"/>
         </Edges>
      </DialogScreen>
   </Graph>
</Request>
```

Figure 4-2 Success response example

```
<?xml version="1.0" encoding="UTF-8"?>
<Response version="1.1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="%API_URL%?api=default&amp;rschema=default">
   <OperationStatus>
      <Code>0</Code>
      <Message>Request Completed</Message>
   </OperationStatus>
   <Content/>
</Response>
```

NOTE: See *Appendix I* possible settings and default values.

4.1.2 Append graph

Table 4-2 Append graph

%API_URL%?api=graph&method=append&parentID={nodeId}			
POST	Description	Appends a new subgraph on target MFP existing workflow graph. If parent node Id is not provided, the new graph will be appended to the root node on the device and will appear as a new menu option when accessing the first EC screen.	
	Payload	IN	XML: Graph
		OUT	If 200, XML response with code 0 If 400, XML response with error code
	Response Code	200 OK – success	
		400 Bad request Error code -1: Product not licensed Error code -3: Error parsing xml payload Error code -5: Id not corresponding to a valid job Error code -8: Unsupported scan settings Error code -10: Device is busy (silent mode) Error code -12: Unexpected error	
		401 Unauthorized access – if basic authentication fails	

Table 4-2 Append graph (continued)

		411 Length required – if content length is not or is badly specified
		500 Internal Server Error – if too many requests are active. Retry recommended.
Schema	Request	&API_URL&?api=graph&schema=append
	Response	&API_URL&?api=common&rschema=default

Request payload example:

NOTE: See the Request payload on the API *"set"* graph method.

Figure 4-3 Append graph, success response example

```
<?xml version="1.0" encoding="UTF-8"?>
<Response version="1.1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="&API_URL&?api=default&amp;rschema=default">
   <OperationStatus>
      <Code>0</Code>
      <Message>Request Completed</Message>
   </OperationStatus>
   <Content/>
</Response>
```

4.1.3 View graph

Table 4-3 View graph

&API_URL&?api=graph&method=view&includeScheduled={boolean}			
GET	Description	Gets the graph from device. The graph fetched corresponds to the graph stored on device at the moment of the api call execution. As a result, it does not include dynamic jobs already executed, neither nodes already scheduled whatever its status is. **includeScheduled(optional)**: Not used in EC1.2.0; its default value is false	
	Payload	IN	—
		OUT	If 200, XML response with the graph data
	Response Code	200 OK – success	
		401 Unauthorized access – if basic authentication fails	
		500 Internal Server Error – if too many requests are active. Retry recommended	
Schema	Request	—	
	Response	&API_URL&?api=graph&rschema=view	

Figure 4-4 View graph, success response example

```
<?xml version="1.0" encoding="UTF-8"?>
<Response version="1.1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="@API_URL@?api=graph&amp;rschema=view">
    <OperationStatus>
        <Code>0</Code>
        <Message>Request Completed</Message>
    </OperationStatus>
    <Content>
        <Graph>
            <MenuOption name="001_TypicalGraph" disableInactivityTimeout="False" id="deaca780-f7cf-
47e1-a3f8-822b7f730615">
                <Description>Select an option</Description>
                <Buttons>
                    <Button action="Back" label="Back"/>
                    <Button action="Exit" label="Exit"/>
                </Buttons>
                <Edges>
                    <RootParent/>
                    <Parent id="7361fa2d-6c6f-4af6-9519-83f0b167e16c"/>
                </Edges>
            </MenuOption>
                :
                :
                {See SetGraph payload}
                :
                :
            <DialogScreen name="Node" disableInactivityTimeout="False" id="7361fa2d-6c6f-4af6-9519-
83f0b167e16c">
                <Description>Message</Description>
                <Buttons>
                    <Button action="Back" label="Back"/>
                    <Button action="Exit" label="Exit"/>
                    <Button action="Next" label="Next"/>
                </Buttons>
                <Message>Document file type: %_Scan.ScanBasicOptions.FileType_%&lt;br/&gt;Scanned
pages: %_Scan.ScanNumPages_%&lt;br/&gt;Scanned files:
%_Scan.FileCount_%&lt;br/&gt;&lt;br/&gt;&lt;WHILE type="process" scope="cycle"
delimiter="&lt;br/&gt;"&gt;  %_Process.Name_%:&lt;br/&gt;     %_Process.Result_% -
%_Process.ResultMessage_%&lt;br/&gt;     (Files:
%_Process.FileCount_%)&lt;br/&gt;&lt;/WHILE&gt;</Message>
                <Edges>
                    <Parent id="2b94b08b-7cdd-46c5-90ea-bd3a3969c92b"/>
                </Edges>
            </DialogScreen>
        </Graph>
    </Content>
</Response>
```

4.1.4 Clear graph

Table 4-4 Clear graph

@API_URL@?api=graph&method=clear&includeScheduled={boolean}			
GET	Description	Clears the full graph from the device. If includeScheduled is not set, this call has no effect on processes already scheduled for execution. includeScheduled(optional): Not supported in EC 1.2.0; its default value is false.	
	Payload	IN	—
		OUT	If 200, XML response with code 0 If 400, XML response with error code
	Response Code	200 OK — success	

Table 4-4 Clear graph (continued)

		400 Bad request Error code -2: Invalid request parameters Error code -10: Device is busy Error code -12: Unexpected error
		401 Unauthorized access – if basic authentication fails
		411 Length required – if content length is not or is badly specified
		500 Internal Server Error – if too many requests are active. Retry recommended.
Schema	Request	—
	Response	%API_URL%?api=common&rschema=default

Figure 4-5 Clear graph, response example

```
<?xml version="1.0" encoding="UTF-8"?>
<Response version="1.1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="%API_URL%?api=common&amp;rschema=default">
   <OperationStatus>
      <Code>0</Code>
      <Message>Request Completed</Message>
   </OperationStatus>
   <Content/>
</Response>
```

4.1.5 Modify node

Table 4-5 Modify node

%API_URL%?api=graph&method=modifyNode&includeScheduled={boolean}			
POST	Description	Modifies a graph node on the target MFP. By including scheduled nodes, they can be modified in order to change their parameters before they are executed. includeScheduled(optional): Not supported in EC 1.2.0; its default value is false.	
	Payload	IN	XML: Node
		OUT	If 200, XML response with code 0 If 400, XML response with error code
	Response Code	200 OK – success	
		400 Bad request Error code -3: Error parsing xml payload Error code -5: Id not corresponding to a valid node Error code -8: Unsupported scan settings. (if it is a scan node) Error code -10: Device is busy Error code -12: Unexpected error	
		401 Unauthorized access – if basic authentication fails	

Table 4-5 Modify node (continued)

		411 Length required – if content length is not or is badly specified
		500 Internal Server Error – if too many requests are active. Retry recommended.
Schema	Request	%API_URL%?api=graph&schema=modifyNode
	Response	%API_URL%?api=common&rschema=default

Figure 4-6 Modify node, request payload example

```
<?xml version="1.0" encoding="UTF-8"?>
<Request version="1.1.0">
   <Form name="Modified Metadata Name" id="fe1253dc-902c-4f49-a0ed-7411001e6b64">
      <Title>Modified Metadata Title</Title>
      <Buttons>
            <SystemButton action="Back" label="Back"/>
            <SystemButton action="Next" label="Next"/>
         </Buttons>
      <FormOptions>
            :
            :
         {Form Options}
            :
            :
         </FormOptions>
   </Form>
</Request>
```

Figure 4-7 Modify node, response example

```
<?xml version="1.0" encoding="UTF-8"?>
<Response version="1.1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="%API_URL%?api=common&amp;rschema=default">
   <OperationStatus>
      <Code>0</Code>
      <Message>Request Completed</Message>
   </OperationStatus>
   <Content/>
</Response>
```

4.1.6 Delete node

Table 4-6 Delete node

%API_URL%?api=graph&method=deleteNode&nodeId={nodeId}&includeScheduled={boolean}

GET	Description	Removes a node from graph. Removing a node will also remove any sub-graph depending directly exclusively on this node. **NOTE:** If not provided includeScheduled is considered false. HP EC 1.2.0 only supports includeScheduled=false.	
	Payload	IN	—
		OUT	If 200, XML response with code 0 If 400, XML response with error code
	Response Code	200 OK – success	
		400 Bad request	

Table 4-6 Delete node (continued)

		Error code -5: Id not corresponding to a valid node Error code -10: Device is busy (silent mode) Error code -12: Unexpected error
		401 Unauthorized access – if basic authentication fails
		411 Length required – if content length is not or is badly specified
		500 Internal Server Error – if too many requests are active. Retry recommended
Schema	Request	—
	Response	%API_URL%?api=common&rschema=default

Figure 4-8 Delete node, response example

```
<?xml version="1.0" encoding="UTF-8"?>
<Response version="1.1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="%API_URL%?api=common&amp;rschema=default">
   <OperationStatus>
      <Code>0</Code>
      <Message>Request Completed</Message>
   </OperationStatus>
   <Content/>
</Response>
```

5 Appendix I: API settings reference

The following completes the information provided in the API XSD documents that are more related to the structure and content type. Depending on the device model, some parameters may vary, and are subject to device specific capabilities outlined in the specifications.

5.1 Navigation settings

NAME	TYPE	REQUIRED (PUT)
LabelA	string	TRUE
LabelB	string	FALSE
JobDescription	string	FALSE
Permanent	boolean	TRUE
HideDeleteButton	boolean	FALSE
HideFileName	boolean	FALSE
OneButton	boolean	FALSE
ShowSummaryScreen	boolean	FALSE

5.2 Scan settings

NAME	TYPE	REQUIRED (PUT)
Type	string	TRUE
Color	string	TRUE
Resolution	string	TRUE
Duplex	boolean	TRUE
Source	string	TRUE
MediaSize	string	TRUE
PageContent	string	FALSE
QualityMode	string	FALSE
Sharpness	integer	FALSE
Darkness	integer	FALSE
BackgroundRemoval	integer	FALSE
Orientation	string	FALSE
Multipage	boolean	FALSE
NumOfPages	integer	FALSE

NAME	POSSIBLE VALUES
Type	“jpg”, “pdf”, “tiff”, “mtiff”, “xps”
Color	“color”, “bw”, “grayscale”
Resolution	“75”, “150”, “200”, “300”, “400”, “600”
Duplex	boolean
Source	“auto”, “adf”, “flatbed”
MediaSize	“auto”, “letter”, “legal”, “exec”, “a3”, “a4”, “a5”, “b5”, “b5_env”, “j_double_postcard”, “dl_env”
PageContent	“text”, “graphic”, “mixed”
QualityMode	“small”, “medium”, “large”
Sharpness	“1”, “2”, “3”, “4”, “5”
Darkness	“0”, “1”, “2”, “3”, “4”, “5”, “6”, “7”, “8”
BackgroundRemoval	“1”, “2”, “3”, “4”, “5”, “6”, “7”, “8”, “9”
Orientation	“portrait”, “landscape”
Multipage	boolean
NumOfPages	integer

5.3 Metadata and Custom options

Metadata and Custom options are both optional. When specifying Custom options, there are a set of possible options. Any other option will raise an exception:

KEY	POSSIBLE VALUES
removeBlankPages	“on”, “off”
duplexEditable	“true”, “false”
userName	String
bitRate	integer
compressionMode	"NotApplicable", "Normal", "High", "G3", "G4", "JpegTiff6", JpegTTN2", "LZW"
autoCrop	"true", "false"

5.4 Notifications

Notification tag is mandatory with a valid type and condition.

Table 5-1 Notification email

NAME	TYPE	REQUIRED (PUT)
Port	Integer	FALSE
DestAddress	String	TRUE

Table 5-1 Notification email (continued)

FromAddress	String	TRUE
Subject	String	FALSE

5.5 Destinations

- Destination tag: obligatory
- Type attribute: mandatory
- Metadata attribute: optional (default value = false)

Destination email:

NAME	TYPE	REQUIRED (PUT)
Port	Integer	FALSE
DestAddress	String	TRUE
FromAddress	String	TRUE
CcAddress	String	FALSE
BccAddress	String	FALSE
Subject	String	FALSE
FileName	String	FALSE
Notification	Boolean	FALSE

Destination FTP:

NAME	TYPE	REQUIRED (PUT)
Address	String	TRUE
Port	Integer	FALSE
Path	String	FALSE
UserName	String	TRUE
Password	String	FALSE
FileName	String	FALSE
MetadataPath	String	FALSE

Destination Network Folder:

NAME	TYPE	REQUIRED (PUT)
Domain	String	TRUE
Path	String	TRUE

UserName	String	FALSE
Password	String	FALSE
FileName	String	FALSE
MetadataPath	String	FALSE

6 Appendix II: Error codes

The following table provides a summary of all possible API error codes returned when something does not work as expected, or in some cases, to inform of a situation needing attention from the user/operator side, possibly requiring a retry of the failed operation. The generic error message indicates the category of the error, but the description is different for each API call, providing more detailed information of each case.

Table 6-1 API Error codes

Error code	Old NSE error codes		Generic description
	Java	C#	
-1	-1	-11	Product not licensed.
-2	-8	-8	Invalid request parameters.
-3	-11	-3	Error parsing request xml payload.
-4	-12	—	The given admin password is invalid. Please update it (non-FutureSmart only)
-5	-5	-5	Id not corresponding to a valid job.
-6	-8	-8	Job settings could not be loaded.
-7	-6	-6	Unexpected error creating zip file.
-8	-	-8	Unsupported scan settings.
-9	-	-	n/a
-10	-7	-7	Device is busy. Please repeat operation within a few seconds.
-11	-8	-8	Unsupported media size value.
-12	-	-	Unexpected error.

(*) Old error codes indicate values that product NSE notified on similar situations. They are provided as a reference for backwards compatibility for programmers adapting client applications for integration with HP Embedded Capture API.