DirectSmile Partner Image request & response handling (XML/Hotfolder based)

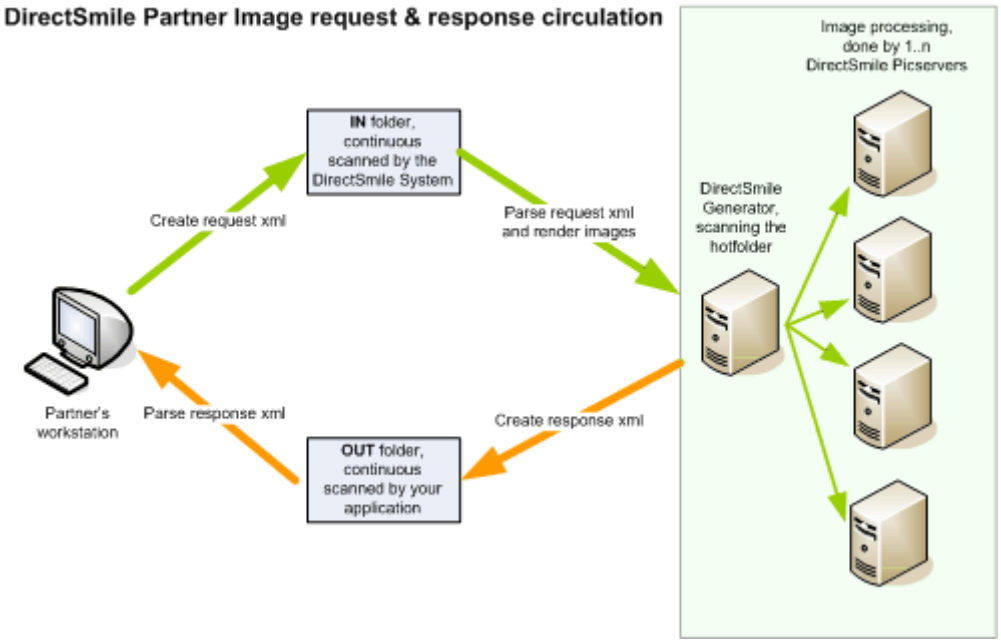
21.11.06

DirectSmile GmbH
Torstraße 164
10115 Berlin

Tel. +49 (0)30 627 77 0
Fax +49 (0)30 622 85 65
www.directsmile.de

General

The hotfolder based Image request gives a DirectSmile partner the opportunity to generate in an easy way images by a DirectSmile System. The communication is xml file based.



Request XML-file

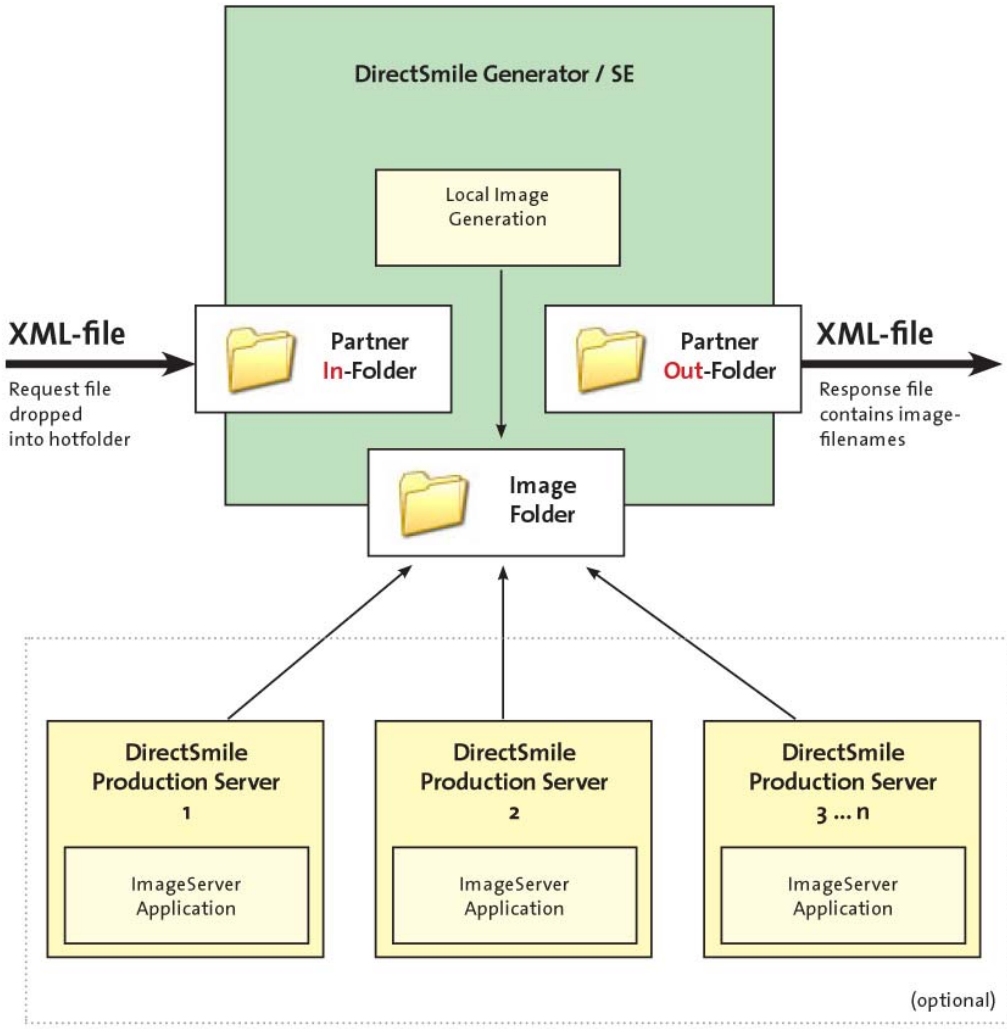
The first file is called the request file and it holds a list of DirectSmile Sets and text for the image personalization of each image that should be processed by the DirectSmile System. The attribute names are case sensitively interpreted.

Response XML-file

The second file is called the response file. This file will be created after all Images are rendered. The response file contains the filenames of all rendered images. If an error occurred while processing the request file, the response may include an error message.

DirectSmile Partner Interface

(Asynchronius operation - For high volume image requests)



Communication is hotfolder based.
Hotfolders don't necessarly have to be on the Generator Computer.

Partner's IN/OUT hotfolder

The hotfolder, where you have to place the request files, is at a fixed location inside the DirectSmile working directory. Usually this folder is located under:

c:\program files\DirectSmile Generator\Partners

The folder contains two subfolders: **IN** and **OUT**. The IN folder is used to place the request files, the OUT folder is used by the DirectSmile System to place the response files.

The path to the application directory of the DirectSmile Generator is stored in the Windows registry and can found in the **GenPath** (REG_SZ) key under the following hive:

HKEY_CURRENT_USER\Software\VB and VBA Program Settings\DirectSmile Generator\System

Please be aware of the fact, that the DirectSmile user is free to define the DirectSmile working directory. Normally the DirectSmile customer runs more then one computer in a DirectSmile environment. So the working directory is usually located on a different server. So please always use the registry entry (see documentation) to find out the proper working directory, or ask your client to define the alternative partner in/out path (see below).

The IN folder is continuously scanned by the DirectSmile System for new request files. The OUT folder must be monitored by your system to get notice of finished jobs by the DirectSmile System.

If you drop your Request file into the IN folder, you have to ensure that the DirectSmile Application is running.
The DirectSmile Generator will load the Request file and delete it afterwards. Then the image processing will start. After rendering all of the requested pictures, the Generator sends a second xml-file to the *\Partners\OUT* folder. This file is called the Response file.
It has the same name as the request file.
So you're application has to monitor the OUT folder to recognize, if the job is done by DirectSmile.
Please keep in mind, that you have to delete the expected response file from the OUT-Folder before sending your Request to the IN folder.

Alternative Partner's IN/OUT hotfolder

If you are not satisfied with the default location of the Partner's Hotfolder you can set an alternative Hotfolder by yourself. The alternative path must be reachable by the DirectSmile Generator.
That's why the alternative Hotfolder must be registered in the DirectSmile Generator.
You can add or edit a path to an alternative Hotfolder in the "General Settings" dialog, which is reachable in the file menu of the Generator main form.

Request files

The request file is XML 1.0 based, the content is UTF-8 encoded. The XML uses XML-Elements and XML-Attributes.

Each request is enclosed in a so called `PartnerRequest` Element. The `PartnerRequest` element contains various attributes, which will be described later. It also contains a collection of `Image` elements, called `Images`.

You can name the file as ever you want, but the extension should be **.xml**. It has to be placed in the IN folder of the partner's Hotfolder (see the "**Hotfolder**" section for more details). The request file will be deleted by the DirectSmile System, after processing the requested images.

The response file, generated by the DirectSmile System has the same title as the request file.

Attributes of the PartnerRequest element

Attribute	Description
JobID="83388"	Optional Your JobID, will be returned in Response-File
OutputFolder="c:\dsmtemp\pics"	Required Output path for the generated images
PartnerID=""	Optional The PartnerID your received from DirectSmile. Please leave empty if you don't have an ID
StatusFilename="c:\test-status.txt"	Optional A filename of a provided status file, where the DirectSmile System may report progress messages of image generation. Refer "Reading Status messages"
LastResponseFilename=""	Optional If you send a Job twice, you can transmit the full path and filename to a former received ResponseFile again. That will avoid a new charging of Smilys.
RedrawAllImages="Off"	Optional If set to "On", all Pictures are been recalculated
ShowSmilyDialog="Off"	Optional If "On", the Generator shows a Smily-dialog and waits for user input. If "Off", the Generator will automatically debit the amount of Smilys, necessary to process all the images.
Silent="Off"	Optional If set to "On", the Generator will not display any progress dialogs.
ImageCroping="Off"	Optional If set to "On", you will receive only the deviating parts of a personalized image. Plus cropping coordinates and a reference to the associated background image and its dimensions.

Elements of the PartnerRequest element

Element	Description
Images	Required List Element contains 1-n Image elements. The Images Element contains no attributes.

The Image element

The Image element specifies one single image request. Usually you will send many different image elements in the Images collection in a request file. In the SetFilename attribute you pass the wanted DirectSmile Set name.
There are two ways to pass the Set name: First, a relative path from the DirectSmile working directory leaded by “\Sets\” –folder. Second, an absolute path to a specific Set folder, that can be anywhere located in your network, for instance on mapped network drive.

Attributes of the Image element

Attribute	Description
ID="1"	Optional ID of the image, will be replaced through a counter if missing.
SetFilename="\Sets\Breakfast\Breakfast.dset”	Required Filename of the DSM-Set (The SetFilename can be relative to Sets-Folder in the working directory of the DirectSmile System, or an absolute path to a specific DirectSmile Set folder.)
TextInImage="Peter"	Required Text to personalize the image with
DPI="300"	Optional Changes the resolution of the resulting image, default is 72
ConvertTo="EPS"	Optional Converts the resulting image (JPG) into passed format. For the moment it converts only to EPS. If you use this attribute, have in mind that it will only convert the images, if the images are newly created. So, delete all the images before you send a new Request with a changed ConvertTo attribute.
Watermark="1"	Optional (0=None, 1=DirectSmile, 2=Generic [Preview]) Default is 0. Places a Watermark on the generated image. If the attribute is set to 1 or 2, no Smilys will be charged for the image and the maximum width is limited to 600 pixel.
Quality="10"	Optional Range 1 to 10. Default is 10. Quality of the resulting JPEG (1=bad, 10=best)
PixelWidth="257"	Optional If set to 0 or the attribute is omitted, the image will be generated in it’s original size. (Valid range 10 to Imagesize -1) Usually used in combination with the Watermark attribute to create web previews.

Watermark="1"	Optional If set to “0” or the attribute is omitted, the image will be generated in it’s original (or determined) size without Watermark. If “Watermark” is set to “1” or “2” a watermark appears on the rendered image. **No Smilys will be debited** and the width or height is limited to 600 pixels. Watermark=”1” Displays a DirectSmile watermark Watermark=“2” Displays a generic watermark.

Request xml sample

```
<?xml version="1.0"?>
<PartnerRequest
      JobID="83388"
      OutputFolder="c:\dsmtemp\pics"
      PartnerID=""
      StatusFilename="c:\test-status.txt"
      LastResponseFilename=""
      RedrawAllImages="Off"
      ShowSmilyDialog="Off"
      Silent="Off"
      ImageCroping="Off">

      <Images>

            <Image
                  ID="1"
                  SetFilename="\Sets\Breakfast\Breakfast.dset"
                  TextInImage="Christoph"
                  DPI="300"
                  ConvertTo="EPS"
            />
            <Image
                  ID="2"
                  SetFilename="\Sets\Balloons\Balloons(1).dSet"
                  TextInImage="Mario"
                  DPI="300"
                  ConvertTo="EPS"
            />
            <Image
                  ID="3"
                  SetFilename="\Sets\Balloons\Balloons(1).dSet"
                  TextInImage="Susann"
                  DPI="300"
                  ConvertTo="EPS"
            />
            <Image
                  ID="4"
                  SetFilename="\Sets\Balloons\Balloons(1).dSet"
                  TextInImage="Peter"
                  DPI="300"
                  ConvertTo="EPS"
            />

      </Images>
</PartnerRequest>
```

Response files

The response file is XML 1.0 based, the content is UTF-8 encoded. The XML uses XML-elements and XML-Attributes.

The response file is basically of the same structure like the request file. But it contains some new elements and attributes.

Response xml sample

```
<?xml version="1.0"?>
<PartnerResponse
      JobID="83388"
      OutputFolder="c:\dsmtemp\pics"
      PartnerID="CREO"
      StatusFilename="c:\test-status.txt"
      LastResponseFilename=""
      UserID="s7k3"
      ImageCroping="Off">

      <Error
            ID="0"
            Message="OK"
      />

      <Images>
            <Image
                  ID="1"
                  FileName="rI32EwWotBcwV+qWZ3t73Yrx6GmslhJ9CwmFasWjhuNw"
                  DPI="300"
                  ConvertTo="EPS"
            />

            <Image
                  ID="2"
                  FileName="rI32EwWotBcwV+qWZ3t73Yrx6GmhlRd4CASFasWjhuNw"
                  DPI="300"
                  ConvertTo="EPS"
            />

            <Image
                  ID="3"
                  FileName="rI32EwWotBcwV+qWZ3t73Yrx6GmkmxV7BQ2FasWjhuNw"
                  DPI="300"
                  ConvertTo="EPS"
            />

            <Image
                  ID="4"
                  FileName="rI32EwWotBcwV+qWZ3t73Yrx6GmnkRd2Dw6FasWjhuNw"
                  DPI="300"
                  ConvertTo="EPS"
            />

      </Images>

</PartnerResponse>
```

The `PartnerResponse` element contains a child called `Error`. If the `Error` attribute `ID` is zero the Job was successfully processed. If `ID` has a value beside 0 an Error has occurred and is mentioned in the `Message` attribute. Depending on an Error you may find the attributes `SetFilename` and `TextInImage` in the responding `Image` element again. For more about errors refer "**Error handling**"

The `Image` element contains a new attribute called `Filename`. This attribute holds the RC4 encrypted filename relative to the mentioned `OutputFolder`. You find more details about decrypting in the "**Decrypting the image filename**" section.

Most important the order of the Images is the same than in your request file. This allows you easily to match the filenames to the images.

Attributes of the PartnerResponse element

Attribute	Description
`JobID="83388"`	Same as in request
`OutputFolder="c:\dsmtemp\pics"`	Same as in request
`PartnerID="CREO"`	Same as in request
`UserID ="s7k3"`	The UserID needed for decrypting the image filename
`ImageCroping="Off"`	Same as in request

Elements of the PartnerResponse element

Element	Description
`Error`	occurred errors
`Images`	List Element contains 1-n Image elements. The Images Element contains no attributes.

The Image element

The response `Image` element contains the RC4 encrypted and Base64 encoded file name of the rendered image. If an error occurred while rendering these image the element may contain three attributes:

1. `ErrorMsg` attribute, describing the error.
2. `SetFilename` attribute, repeating the requested DirectSmile Set
3. `TextInImage` attribute, repeating the requested text

If in the `Cropping` attribute was the value “ON” passed in the request file, the `Image` element will contain a `CroppingRect` element.

Attributes of the Image element

Attribute	Description
`ID="1"`	Sequence index.
`FileName="DSM_123456000.JPG"`	Filename of the rendered image. (If a Partner ID was submitted, this text is RC4 crypted and Base64 encoded.)
`ErrorMsg="File not found [Sets\Balloons\Balloons.dset]"`	Error message, only if an error occurred.
`SetFilename="Sets\Balloons\Balloons.dset"`	Requested DirectSmile Set, only if an error occurred.
`TextInImage="Susann"`	Requested text to personalize, only if an error occurred.
`DPI="300"`	Image resolution, only if the attribute was passed in the request file.

Decrypting the image filename
(Only required if you submit a partner ID)

One `Image` element in the response file contains the encrypted and Base64 encoded filename and looks like:

```
<Image
      ID="1"
      FileName="rI32EwWotBcwV+qWZ3t73Yrx6GmslhJ9CwmFasWjhuNw"
/>
```

Before you can decrypt the image filename, you have to decode the string from it's Base64 encoding.
To decrypt the image filename you'll need a standard RC4 algorithm. If you don't have one, don't hesitate to call us, we will send you some sample code.
The decryption key consists of two parts:

1. Your PartnerKey (You should have got it by DirectSmile)
2. The UserID (You'll find this key in the `UserID` attribute of the `PartnerResponse` element)

```
<PartnerResponse
      ...
      UserID="s7k3"
      ...
```

If you put these too keys together (ImageKey = PartnerKey + UserID) you have the final key to encrypt the filename via a RC4-Algorythm.

After decrypting the filename, You should have a string like `"DSM_477749000.JPG".` Just add your `OutputFolder` before the filename and you're done.

```
<PartnerResponse
      ...
      OutputFolder="c:\dsmtemp\pics"
      ...
```

Finally the filename for the Image no. 1 is:

```
"c:\dsmtemp\pics"  + "\" +  "DSM_477749000.JPG" =
"c:\dsmtemp\pics\DSM_477749000.JPG"
```

Just do this for every image in list. The sequence of the images is identical to the sequence in the request file you sent.

Elements of the Image element

Element	Description
CroppingRect	If Cropping was set to “ON” each Image element will contain a CroppingRect.

Attributes of the CroppingRect element

Attribute	Description
cLeft="10"	Left span in pixel, depending on the background image.
cTop="10"	Top span in pixel, depending on the background image.
cHeight="100"	Height of the resulting cropped image in pixel.
cWidth="100"	Width of the resulting cropped image in pixel.
BackgroundImageFilename="Y:\Sets\Balloons\bck.tif"	Path to the used background image.
BackgroundHeight="2500"	The height of the background image in pixel.
BackgroundWidth="1400"	The width of the background image in pixel.

The Error element

The `Error` element gives a fast overview if any errors occurred while processing the request. Although each error will be part the specific `Image` , where the error occurred. The general `Error` element will hold the last error if more than one occurred. For more about errors refer "**Error handling**")

Attributes of the Error element

Attribute	Description
`ID="1"`	`If the ID differs from 0 (zero) an error occurred.`
`Message="Set not found [\Sets\foo\foo.dset]"`	`Distinct list of all error messages occurred, separated by semicolon.`

Error handling

Basically there are two places where an error is indicated in the response file.

1. The `Error` element (Child of `PartnerResponse`)
2. The `ErrorMsg` attribute in an `Image` element

Two explain these two types of errors we will show you 3 simple examples

Example 1: Invalid OutputFolder

If the specified `OutputFolder` isn't valid the image can't be stored. The job is interrupted at a very early stage. The response file could look like this:

```
<PartnerResponse
      JobID="83388"
      OutputFolder="c:\dsmtempx"
      ...

      <Error
            ID="-1"
            Message="Invalid output folder"
      />

      <Images/>

</PartnerResponse>
```

The `ID` attribute is set to "-1" to indicate that an error occurred. The `Message` attribute specifies the error. The `Images` collection is empty.

Example 2: Smily Error
If the DirectSmile System was unable debit the Smilys needed to process the requested images the job is interrupted at a very early stage as well. The response file could look like this:

```
<PartnerResponse
      JobID="83388"
      OutputFolder="c:\dsmtempx"
      ...
      <Error
            ID="-1"
            Message="Smilys could not be debited"
      />

      <Images/>

</PartnerResponse>
```

Example 3: Invalid Set Name
Let's say you sent a request with 2 different images. Each of the image should be rendered with a different DirectSmile Set. But one of the Sets is not available on the system.

Request:

```
<Image
      ID="1"
      SetFilename="Sets\Balloons\Balloons.dset"
      TextInImage="Susann"
/>

<Image
      ID="2"
      SetFilename="\Sets\Breakfast\Breakfast.dset"
      TextInImage="Susann"
/>
```

Response:

```
<PartnerResponse
      JobID="83388"
      OutputFolder="c:\dsmtempx"
      ...

      <Error
            ID="-1"
            Message="File not found [Sets\Balloons\Balloons.dset]""
      />

      <Images/>

      <Image
            ID="1"
            ErrorMsg="File not found [Sets\Balloons\Balloons.dset]"
            SetFilename="Sets\Balloons\Balloons.dset"
            TextInImage="Susann"
      />
      <Image
            ID="2"
            FileName="rI32EwWotBcwV+qWZ3t73Yrx6GmslhJ9CwmFasWjhuNw"
      />
```

The Set for Image 1 could not be found. So an error is indicated as well in the `Error` element as in the `Image` itself.

Response xml sample (including an error)

```
<?xml version="1.0"?>
<PartnerResponse
      JobID="83388"
      OutputFolder="c:\dsmtemp\pics"
      PartnerID="CREO"
      StatusFilename="c:\test-status.txt"
      ReceiptFilename=""
      UserID="s7k3"
      ImageCroping="Off">

      <Error
            ID="0"
            Message="OK"
      />

      <Images>
            <Image
                  ID="1"
                  FileName="DSM_6276327000.JPG"
            />

            <Image
                  ID="2"
                  FileName="DSM_6272347000.JPG"
            />

            <Image
                  ID="3"
                  FileName="DSM_1236327000.JPG"
            />

            <Image
                  ID="4"
                  FileName="DSM_5343465000.JPG"
            />

      </Images>

</PartnerResponse>
```

Reading Status messages

The DirectSmile System can provide you with status messages while processing you request. The format is xml based as well. If you like to retrieve status messages by the DirectSmile you have to pass a filename in the `StatusFilename` attribute in your request file.

The status file is XML 1.0 based, the content is UTF-8 encoded. The XML uses XML-elements and XML-Attributes.

The status file has one single element, the `Status`.

Attributes of the Status element

Attribute	Description
`ID="1"`	Internal status code.
`Percent="100"`	How much is processed in percent.
`Description ="finished"`	Human readable status message.
`CurrentIndex="12"`	Row-number that is currently rendering
`CurrentText="Peter"`	Text in image that is currently rendering

Creating On-the-fly Sets

Instead of pointing to a existing Set in your Set library, you can let the System create a Set instantly while generating the image. This may be helpful if you plan to combine the DirectSmile personalization and various or often changing background images. For example, place some personalized seagulls on your beach photos, you took at your last summer holiday.

All you have to do is to tell the System where it will find your background image, the coordinates, the height and width of the frame where the personalization will be placed and finally the name of the predefined Style sheet, the personalization should base upon. These attributes will be added to the `Image` element in the PartnerRequest xml.

The left and top position, the width and height of the personalization frame must be specified in pixel based on a point of origin in the upper left corner of the background image.

If you want to use the On-the-fly Set creation the `SetFilename` attribute must be empty. If the attribute has a value the System will always try to load the Set first and ignore the additional attributes.

The Background image must be reachable for the Generator and all Picservers involved in the DirectSmile System. While the Set creation new files can be created temporarily by one of the computers, therefore it is recommended to guarantee the permission to write and change files.

Additional Attributes for an On-the-fly Set creation

Attribute	Description
SetFilename=""	Leave this attribute empty to indicate an On-the-fly Set request.
BackgroundImageFilename="x:\IMG_0383.jpg"	Required Full path to the variable background image. Supported file formats are JPEG and TIFF in RGB.
NewFrameWidth="300"	Required Width in pixel of the personalization frame.
NewFrameHeight="100"	Required Height in pixel of the personalization frame.
NewFrameLeft="20"	Required Left position in pixel of the personalization frame.
NewFrameTop="20"	Required Top position in pixel of the personalization frame.
StyleSheet="Clouds"	Required Name of your predefined Style Sheet. The Style sheet must be available in the Style sheet list of the Generator.