

# Using *-extended-features*

---

This technical note describes how to access a feature defined in HP Output Server that is present in the printer's capability ticket but has no corresponding HP Output Server attribute.

The `-extended-features` attributes facilitates accessing a feature present in the printer. These features do not have corresponding attributes in the `mapper.xml` file which is normally used to map attributes to the features.

A capability ticket is basically needed to perform a job by using the printer's features effectively. The `-extended-features` attributes need to have a capability ticket as a prerequisite before any command for the attributes get executed.

## **-extended-features syntax**

`-extended-features` uses the following syntax:

```
pdpr -dlogical_destination -x"-extended-features '<pwg:OrientationRequested
xmlns:pwg=\"http://www.hp.com/schemas/imaging/con/pwg/sm/1.0/\">Landscape</pwg:
OrientationRequested >'<filename>
```

**Table 1. -extended-features command and description**

Command	Description
pdpr	Print command
-d<logical_destination>	-d option Logical destination name
-x	Option to specify the feature attribute
-extended-features	Feature attribute
`-extended-features' '<pwg:OrientationRequested xmlns:pwg=\"http://www.hp.com/schemas/imaging/con/pwg/sm/1.0/\">Landscape</pwg:OrientationRequested >'	XML extract from the capability ticket.
filename	The document to be printed

### Using option files

These commands are case-sensitive. Ensure that single quotes and spaces do not get omitted.

To access a particular feature unavailable in HP Output Server attributes, the `set -extended-features` command is used followed by the `pdpr` command for sending the job.

For example, to access the one-angled staple option that is unavailable in HP Output Server, use the `-extended-features` attribute as follows:

```
pdpr -dlogicalDestination -x"-extended-features '  
  <Features:QQQStapleOptionxmlns:Features=\"http://www.hp.com/schemas/i  
  maging/con/dictionaries/1.0/\">ONEANGLED</Features:QQQStapleOption>' "  
  multiPageDocument
```

The value of `-extended-features` is an XML fragment that describes the desired feature and its value from the destination's capabilities ticket.

---

**NOTE:** If `-extended-features` is set at a job and it conflicts with `-extended-features` set at the destination, you may experience unexpected results.

---

## Using option files

Option files can be used to make delivery with the `pdpr` command. The use of option files makes the execution of `pdpr` command easy and uncomplicated. It helps to set one or more attributes. Option files also help to implement multiple attributes at once and contain one or more attribute specifications separated by spaces.

Option files are used to implement one or more extended features at a time.

To write an option file, perform the following steps:

1. List the features and options which need to be applied for the print job.
2. Open Notepad.
3. Write the extended feature for each of the feature option combinations as shown below. This example targets two feature option combinations.

a. Feature 1

```
-extended-features '<Features:QQQStapleOptionxmlns:Features=\"http://www.  
  hp.com/schemas/imaging/con/dictionaries/1.0/\">ONEANGLED</Features:QQ  
  QStapleOption>'
```

b. Feature 2

```
-extended-features '<pwg:OrientationRequestedxmlns:pwg=\"http://www.hp.co  
  m/schemas/imaging/con/pwg/sm/1.0/\">Landscape</pwg:OrientationRequest  
  ed>' -extended-features '<Features:QQQStapleOptionxmlns:Features=\"http  
  ://www.hp.com/schemas/imaging/con/dictionaries/1.0/\">DEFAULT</Featur  
  es:QQQStapleOption>'
```

4. To use an option file attribute with the `pdpr` command:

```
pdpr -d<printerName> -o<optionsFileName> <jobName> command to deliver the  
  document.
```

The document gets delivered.

## -extended-features requirements

An argument for -extended-features has the following requirements:

- The argument must come from the capabilities ticket. It must be a feature and option combination available in the capabilities ticket.
- Make sure that there is an invocation string for the option. An argument works only if there is an invocation string available.
- Qualify the feature with a namespace.

The feature and option must not map to a similar feature that HP Output Server supports, for example, the orientation feature.

## Steps to extract the XML fragment used for -extended-features

Before you perform the steps to extract the XML fragment used for -extended-features attributes, ensure that:

- '<feature>value</feature>' format is used to provide input for -extended-features.
- For multiple values, the following format is used:
  - a. -extended-features '<feature>value</feature>', '<featureTwo>value</featureTwo>'. Each value is enclosed in single quotes and one space is provided between each of the values.
  - b. The namespace is specified in a feature as: <featurexmlns="namespace\">

To extract the XML fragment used for -extended-features attributes, perform the steps given below:

Example, using -extended-features to access the angled staple option.

1. Identify the desired feature in the capabilities ticket.

```
<Featureid="QQQStapleOption" type="Exclusive" display="true" intentRequired="false" extensionAllowed="true" feature-NS="Features" feature-Type="QQQStapleOption">
```

2. Locate the option within the QQQStapleOption feature that matches the specific value for the desired stapling option.

```
<Option
constraintRefs="Features_MediaSizeName_a5
_Features_QQQStapleOptionFeatures_MediaSizeName_executive_Features_QQQStapleOptionFeatures_MediaSizeName_jis_b5_Features_QQQStapleOptionFeatures_MediaSizeName_jis_exec_216x330mm_Features_QQQStapleOptionFeatures_MediaSizeName_legal_Features_QQQStapleOptionFeatures_MediaSizeName_prc_16k_Features_QQQStapleOption">
<Features:QQQStapleOption>ONEANGLED</Features:QQQStapleOption>
<Invocations>
<InvocationprotocolRef="PJL" sequence="110"><![CDATA[@PJLSET FINISH=STAPLE]]>
</Invocation><InvocationprotocolRef="PJL" sequence="111"><![CDATA[@PJLSET STAPLEOPTION=ONEANGLED]]>
</Invocation>
</Invocations><DisplayStringValue="StapleOneAngled" lang="en"/></Option>
```

### -extended-features restrictions

3. Extract the following XML fragment for the desired option:

```
<Features:QQQStapleOption>ONEANGLED)
</Features:QQQStapleOption>
```

4. Add the namespace. To locate the correct namespace, note the feature-NS element in the feature header. In this example the namespace is Features. At the top of the capabilities ticket is a capabilities element that defines all of the namespaces.

#### Example

```
<Capabilities
xmlns="http://www.hp.com/schemas/imaging/con/capabilities/1.1/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:pwg="http://www.hp.com/schemas/imaging/con/pwg/sm/0.95/"
xmlns:Features="http://www.hp.com/schemas/imaging/con/dictionaries/1.0/"
capabilitiesName="HP 9000 Printer Capabilities">
```

The namespace for xmlns:Features is defined as follows:

```
xmlns:Features="http://www.hp.com/schemas/imaging/con/dictionaries/1.0/"
```

---

NOTE: The namespace is added after the declaration of the feature. Also, because command shells interpret quotation marks differently, quotation marks within the XML fragment must be escaped.

---

5. Use the pdpr command to deliver the job.

```
pdpr -dlogicalDestination
-x"-extended-features '<Features:QQQStapleOption
xmlns:Features="http://www.hp.com/schemas
imaging/con/dictionaries/1.0/">ONEANGLED/Features:QQQStapleOption' "
multiPageDocument
```

---

NOTE: Setting -extended-features requires knowledge of XML and familiarity with the physical destination's capabilities ticket. To gather more information for using this attribute, see "About mapper.xml file", on page 5.

---

## -extended-features restrictions

The following restriction needs to be adhered to at the time of using -extended-features:

- If a feature is included in the mapper.xml file, a corresponding HP Output Server attribute exists. In this case, do not use -extended-features to specify features that are already defined in the mapper.xml file. Do not use -extended-features to select HP Output Server attributes.

For example, if the media is selected through -extended-features for a text document being sent to a PostScript device, the text-to-PostScript transformer cannot access this information because the transformer cannot read -extended-features. The transformer overrides the command in -extended-features during the text-to-PostScript transformation.

## About mapper.xml file

HP Output Server utilizes attributes and values to express intent while the destination capabilities are expressed through features and values. Mapping helps to translate the attributes into their corresponding features so that the DSM server can properly process the constraints and the invocation strings. Mapping is performed by the DSM server which uses a file named `mapper.xml` that exists in the `/etc` subdirectory of the HP Output Server installation. This text file enumerates the mappings between the attributes and the features. It contains one-to-one mappings or multiple-to-one mappings in both directions.

The use of mapping helps to simplify the use of advanced features by allowing you to specify options through standard HP Output Server attributes and values. Otherwise, you would have to constantly specify the XML fragments by using the `-extended-features` attribute. The use of attributes, however, requires an update to both `mapper.xml` file and DPAOIDs. So an advanced feature support through attributes is not always immediately available. The `-extended-features` attribute needs to be utilized until the next update to these files is available.

## Mapping of attributes to the features

Mapping expresses the relationship between attributes and features. Mappings are bi-directional. There are different ways of expressing mappings between the features and attributes.

The following examples illustrate the different ways to express mapping:

### 1. Simple feature <--> attribute mapping

In this style of mapping, one HP Output Server attribute is mapped to exactly one feature in the cap ticket.

Examples

- ```
<class name="copycount">
  <attributename="copies" type="cardinal"/>
  <feature namespace="pwg" name="Copies" type="integer"/>
  <mapping>
    <equivalence>
      <!--
      The equivalence element under mapping indicates that the attribute and
      the feature have the same meaning in the different models. In order to
      set one from the other, you just use the value without modification.
      -->
      <attribute-ref name="copies"/>
      <feature-ref name="Copies"/>
    </equivalence>
  </mapping>
</class>
```
- ```
<class>
  <!--
  Note that in this class the feature and attribute both have units. Since
  the units are not the same, there is an implied conversion. If a
  conversion element is found whose units match the units of the attributes
  and features, it will be used. If a conversion is not found, a straight
  assignment will be done.
```

About mapper.xml file

```
-->
<attribute name="x-image-shift" type="integer" units="point"/>
<feature namespace="pwg" name="XImageShift" type="integer"
units="hundredths_of_mm"/>
<mapping>
<equivalence>
<!--
Rename feature and attribute to feature-ref and attribute-ref in the
context of mapping
-->
<attribute-ref name="x-image-shift"/>
<feature-ref name="XImageShift"/>
</equivalence>
</mapping>
</class>
```

- `<class>`

```
<attribute name="input-tray-select" type="text"/>
<feature namespace="pwg" name="inputBin" type="text"/>
<mapping>
```

```
<!--
```

The translation element under mapping indicates that while the attributes and features represent the same concept, they have a different way of expressing the value. The translation element maps the domain of the attribute to the domain of the feature. There may be many translation elements, depending on the number of domain values.

```
-->
```

```
<translation>
<attribute-ref name="input-tray-select" value="envelope"/>
<feature-ref name="inputBin" value="EnvFeed"/>
</translation>
</mapping>
<mapping>
<translation>
<attribute-ref name="input-tray-select" value="tray1"/>
<feature-ref name="inputBin" value="Tray1"/>
</translation>
</mapping>
<mapping>
<translation>
<attribute-ref name="input-tray-select" value="tray2"/>
```

```

<feature-ref name="inputBin" value="Tray2"/>
</translation>
</mapping>
<mapping>
<translation default="true">
<!--
The default attribute of the translation element indicates that this
translation is for default values. This is an optional construct that
allows for the setting of default values of an exact translation when one
cannot be found. At least one of either a feature element or an attribute
element must be present. If one of the elements is not present, there is
no default value for that element.
-->
<attribute-ref value="default-attribute--value"/>
<feature-ref value="default-feature-value"/>
</translation>
</mapping>
</class>

```

## 2. Simple 1 feature <--> n attributes mapping

In this style of mapping, one HP Output Server attribute is mapped to 'n' number of features in the cap ticket.

Examples

- <class>

```

<!--
This is a mapping from 1 feature to 2 attributes
-->
<attribute name="sides"/>
<attribute name="plex"/>
<feature namespace="pwg" name="sides"/>
<mapping>
<!--
The mapping is a translation with the specified translations. Entries
should also be added for those cases that may not make sense or are
possibly for completeness. If an attribute is left out, it will not be set
during the translation. The most specific match will be the match that is
used.
-->
<translation>
<attribute-ref name="sides" value="1"/>
<attribute-ref name="plex" value="simplex" optional="true"/>
<feature-ref name="sides" value="OneSided"/>

```

About mapper.xml file

```
</translation>
</mapping>
<mapping>
<translation>
<attribute-ref name="sides" value="2"/>
<attribute-ref name="plex" value="duplex" optional="true"/>
<feature-ref name="sides" value="TwoSidedLongEdge"/>
</translation>
</mapping>
<mapping>
<translation>
<attribute-ref name="sides" value="2"/>
<attribute-ref name="plex" value="tumble"/>
<feature-ref name="sides" value="TwoSidedShortEdge"/>
</translation>
</mapping>
</class>
```

### 3. Complex feature <-> attribute mapping

In this style of mapping, the sub features present in the attributes of HP Output Server are mapped to the features in the cap ticket.

Examples

- `<class>`

```
<!--
This is a complex mapping from feature to attribute. The value of the
feature must be parsed to determine if it will set one or two attributes.
Additionally, the feature value comes from a sub-feature of the feature.
-->
<attribute name="input-document-format"/>
<attribute name="character-set" optional="true"/>
<feature namespace="features" name="ContentFormat" type="composite">
<sub-feature namespace="pwg" name="DocumentFormat" type="regex">
<!--
Sub-features indicate that the feature is a structure and this is the way
to address the value of a particular field in the feature.
-->
<regex pattern="(.*)(; charset=){0,1}(.*)" matchcount="2" >
<!--
```

The regex pattern identifies another pattern that should be used to break apart the base pattern. The regex pattern should match the base pattern since there is only one regex allowed per sub-feature. The match elements assign symbolic names to the matched parts so they can be referenced later.

```
-->
<match name="format" regex-ref="1"/>
<match name="charset" regex-ref="3" optional="true"/>
</regex>
</sub-feature>
<composition>
<!--
This composition element tells how to construct the value of the feature
based on the attribute values.
-->
<strcat value-of="format"/>
<conditional value-of="charset">
<!--
The conditional value indicates that this part of the expression is
dependent on the value-of section. It is an implied reference to an
attribute.
-->
<strcat value="; charset="/>
<strcat value-of="charset"/>
</conditional>
</composition>
</feature>
<mapping name="pcl" >
<!--
This mapping is only a translation. The value of the sub-feature format
and the attribute represent the same thing in the different domains.
-->
<translation>
<attribute-ref name="input-document-format" value="hp-pcl" />
<feature-ref name="ContentFormat">
<sub-feature-ref name="format" value="application/vnd.hp-PCL" />
</feature-ref>
</translation>
</mapping>
<mapping name="text/iso-885901">
```

### About mapper.xml file

```
<!--  
This mapping is a combination of a translation and an equivalency. As  
above, the sub-feature format and the attribute input-document-format  
represent the same thing in the different domains, so a translation is  
needed. The value of the sub-feature charset and the attribute  
character-set represent the same concept and the values are equivalent.  
-->  
<equivalence>  
<attribute-ref name="character-set"/>  
<feature-ref name="ContentFormat">  
<sub-feature-ref name="charset"/>  
</feature-ref>  
</equivalence>  
<translation>  
<attribute-ref name="input-document-format" value="text"/>  
<feature-ref name="ContentFormat">  
<sub-feature-ref name="format" value="text/plain"/>  
</feature-ref>  
</translation>  
</mapping>  
</class>
```

- `<class>`

```
<!--  
This class maps from one feature with two sub-features to two attributes.  
-->  
<attribute name="printer-x-resolutions-requested" units="dpi"  
type="integer"/>  
<attribute name="printer-y-resolutions-requested" units="dpi"  
type="integer"/>  
<feature namespace="pwg" name="PrinterResolution">  
<sub-feature name="CrossFeedDir" type="integer">  
<!--  
The units are specified as an element here since the units come from a  
different sub-feature of the feature. The value-of attribute of the  
element indicates where to find the values.  
-->  
<units value-of="PrinterResolution/Units"/>  
</sub-feature>  
<sub-feature name="FeedDir" type="integer">  
<units value-of="PrinterResolution/Units"/>
```

```
</sub-feature>
<sub-feature name="Units" type="text"/>
</feature>
<!--
The units of the attributes are fixed so they can be added as attributes
to the attribute element.
-->
<mapping>
<!--
This is a single mapping where the mapping types are equivalence and
translation. The translation is provided to give a default value for the
sub-feature units when converting from attributes to features. Since the
units may not match a conversion may need to be done.
-->
<equivalence>
<attribute-ref name="printer-x-resolutions-requested"/>
<feature-ref name="PrinterResolution">
<sub-feature-ref name="CrossFeedDir"/>
</feature-ref>
</equivalence>
<equivalence>
<attribute-ref name="printer-y-resolutions-requested"/>
<feature-ref name="PrinterResolution">
<sub-feature-ref name="FeedDir"/>
</feature-ref>
</equivalence>
<translation default="true">
<feature-ref name="PrinterResolution">
<sub-feature-ref name="Units" value="dpi"/>
</feature-ref>
</translation>
</mapping>
</class>
```

