May 26, 2004
Candidate Standard 5101.4-2004

The Printer Working Group

Universal Printer Description Format

Version 1.0

Status: Approved

Abstract: This document describes the concept of a Universal Printer Description Format and the set of schemas it is based on. The schemas describe input for a driver/client to assemble general information about the device and its features, to be used in user interfaces or for printing.

This document is a PWG Candidate Standard. For a definition of a "PWG Candidate Standard", see:
ftp://ftp.pwg.org/pub/pwg/general/pwg-process20.pdf
This document is available electronically at:
ftp://ftp.pwg.org/pub/pwg/candidates/cs-upd10-20040526-5101.4.pdf

Title: Universal Printer Description Format

The IEEE-ISTO and the Printer Working Group DISCLAIM ANY AND ALL WARRANTIES, WHETHER EXPRESS OR IMPLIED INCLUDING (WITHOUT LIMITATION) ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

The Printer Working Group, a program of the IEEE-ISTO, reserves the right to make changes to the document without further notice. The document may be updated, replaced or made obsolete by other documents at any time.

The IEEE-ISTO and the Printer Working Group, a program of the IEEE-ISTO take no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights.

The IEEE-ISTO and the Printer Working Group, a program of the IEEE-ISTO invite any interested party to bring to its attention any copyrights, patents, or patent applications, or other proprietary rights, which may cover technology that may be required to implement the contents of this document. The IEEE-ISTO and its programs shall not be responsible for identifying patents for which a license may be required by a document and/or IEEE-ISTO Industry Group Standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention. Inquiries may be submitted to the IEEE-ISTO by e-mail at:

info@ieee-isto.org

The Printer Working Group acknowledges that the IEEE-ISTO (acting itself or through its designees) is, and shall at all times, be the sole entity that may authorize the use of certification marks, trademarks, or other special designations to indicate compliance with these materials.

Use of this document is wholly voluntary. The existence of this document does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to its scope.

About the IEEE-ISTO

The IEEE-ISTO is a not-for-profit corporation offering industry groups an innovative and flexible operational forum and support services. The IEEE Industry Standards and Technology Organization member organizations include printer manufacturers, print server developers, operating system providers, network operating systems providers, network connectivity vendors, and print management application developers. The IEEE-ISTO provides a forum not only to develop standards, but also to facilitate activities that support the implementation and acceptance of standards in the marketplace. The organization is affiliated with the IEEE (http://www.ieee.org/) and the IEEE Standards Association (http://standards.ieee.org/).

For additional information regarding the IEEE-ISTO and its industry programs visit:
http://www.ieee-isto.org.

About the Printer Working Group

The Printer Working Group (or PWG) is a Program of the IEEE-ISTO. All references to the PWG in this document implicitly mean “The Printer Working Group, a Program of the IEEE ISTO.” The PWG is chartered to make printers and the applications and operating systems supporting them work together better. In order to meet this objective, the PWG will document the results of their work as open standards that define print related protocols, interfaces, data models, procedures and conventions. Printer manufacturers and vendors of printer related software would benefit from the interoperability provided by voluntary conformance to these standards.

In general, a PWG standard is a specification that is stable, well understood, and is technically competent, has multiple, independent and interoperable implementations with substantial operational experience, and enjoys significant public support.

Contact information:

The Printer Working Group
c/o The IEEE Industry Standards and Technology Organization
445 Hoes Lane
Piscataway, NJ 08854
USA

UPD Web Page: http://www.pwg.org/updf UPD Mailing List: upd@pwg.org

Instructions for subscribing to the UPD mailing list can be found at the following link:
http://www.pwg.org/mailhelp.html
Members of the PWG and interested parties are encouraged to join the PWG and UPD WG mailing lists in order to participate in discussions, clarifications and review of the WG product.

Table of Contents

The Idea

Printer drivers and clients are facing a number of tasks: Announcing information on printer features to the operating system and the applications running, offering a user interface to enable the user to change driver and device settings, assembling the print job, listening to feedback coming back from the device.
Many of these tasks are repetitive and very similar even for completely different peripherals.
The basic idea for this concept was to develop a standardized method for as much information as possible.

Terminology

Some terms are re-used over and over. So we considered it a good opportunity to introduce them here.

Device Configuration	The driver entry point, lists all the modules of the device description
Optional Unit Configuration	Similar to a device description, but only lists all the modules of a single optional unit
Unit Description	The technical description of one unit. That can be a device or an optional unit for a device
Device Description	A special unit description – the one for the device or base unit
Locale	The collection of user interface text strings and defaults for one locale = natural language
Command Sequence instance	The JCL and PDL stuff referred to by events and parameters
User Policy	The optional instance a system administrator can add to the device configuration to further describe the driver configuration for a single user or user group
Features	The set of attributes a driver can have a printer perform outside the rendering task
Objects	The print objects on a page
Dependencies	Conditions determining relations between features in a user interface. Sometimes called constraints.
Events	A list of key locations in a print job (e.g. job start, document end)
User interface	The user interface of a driver/client

Table 2 – Terminology

The driver's perspective

In today's world the driver model ranges from a true monolithic concept to scripting. System programmers often have to implement a unique driver for each operating system (OS) for each printer the programmer wants to support. The graphics interfaces of each OS, while similar, tend to be different enough to require major re-write of system code for each operating system. In order to support a wide variety of operating systems developers must spend a considerable amount of programmer-months duplicating effort to support functionality for one printer to print on each operating system. Since there is not an infinite amount of resources for programming, printer drivers are not full-featured under all operating systems. Usually the operating system with the largest market share will offer the most functionality. The amount of effort invested in other operating systems decreases proportionate to their market share. A side effect of having to write unique printer drivers is that they tend to become inconsistent across different operating systems. Printer drivers today show different user interfaces and behave differently during a print job. There are valid reasons for drivers to be slightly different due to the operating system, however a lot of the differences are "just because we can."

Another issue with the current driver model is that it is very static. There is no dynamic discovery mechanism for changes in the printer after the initial installation. If accessories are added after the initial installation of a printer, a user must manually install new software into the OS that enables the accessory to be used.

In order for different operating systems to support any printer, there must be a separation of the OS specific components and the printer feature components. This specification defines a printer description file format that enables a printer driver to configure itself based on the unique printer characteristics. The goal is to design or specify a description format that can be used by multiple operating systems.

So the first goal was to choose a file format, which is supported worldwide and on all platforms. The choice is XML with its schemas and instances.

Reasons are apparent.

XML is a well developed standard controlled by the W3C.

XML is human readable and supported by every major platform.

There is a wide range of standard applications to be used to edit XML files. And one of the major advantages of those applications is the validation of schemas and instances.

XMLSpy and XML Authority are just two samples we deployed during the development of this standard. However there are lots more.

A high level overview

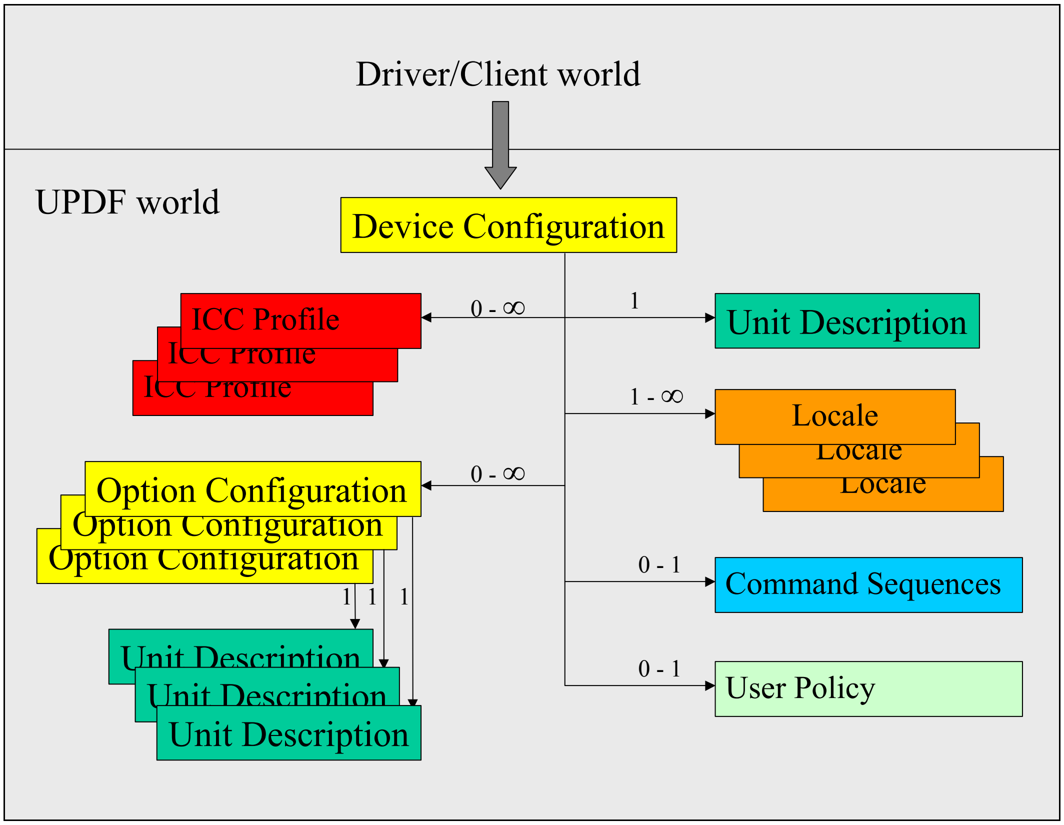

Figure 1: UPDF architecture

Device Configuration
The entry point for a driver or client into the UPDF world is the device configuration file based on the device configuration schema.
It tells about all the components of the actual device description.

Unit Description (the single one in the upper right)
This is the heart of the device description. The instance describes all relevant components of the device.

Locale
A device description needs one or more natural languages to provide the text elements for the user interface.

Command Sequences
A device description normally needs a file, which holds all command sequences to be sent to the device. It was decided to split that from the general technical description stored in the master unit description.

User Policy
A system administrator or similar operator has the chance to add one user policy.

Here user interface modifications dedicated to single users or user groups can be managed up to a certain degree.
Vision a subdirectory for each user or user group, which is supposed to use a special driver configuration. The differences between the subdirectories would be the user policy instance.

ICC Profile
It is possible to assign multiple ICC profiles to a device description.

Option Configuration
An option configuration is very similar to a device description. It describes a single optional unit (e.g. an optional duplex unit).
It cannot live by itself, but is meant to be connected to a device configuration.
An option configuration cannot have its own user policy. A user policy is supposed to be developed after all optional units have been assigned.
<Figure 1> only shows a reference to unit descriptions. However this references represents a unit description, locales, an optional command sequence instance and optional ICC profiles.

Conclusion
The idea is to mirror the real world within the UPDF world. Therefore the pure model as well as each optional unit can be described by a separate unit description instance.
The device configuration instance tells which optional units are currently attached to the device. The IHV is supposed to provide the default device configuration
When the device configuration changes, the driver can react by changing the device configuration file by updating the references to optional units.
Same with the natural languages. If a new locale instance should be added to the configuration or an old one removed, this can be done by updating the device configuration instance.
Keep in mind: The device configuration is not meant to be used to save the currently selected setting (like for a language).
All files other than the device configuration are to be considered read-only.

By interpreting the device configuration the driver/client gets the high level description of the configuration. The heart of the device description are unit description instances, used to describe the pure printer as well as the accessories. This is the technical description of all the features and objects a driver/client is interested in (as much as covered by this specification).

Text strings to be used in user interfaces and PDL specific command sequences are kept out of the unit device description. They are stored in separate locale and command sequences instances.

This is a rough overview about the information you will find in this document. We hope we’ve set you to the right starting point.

Major components of a unit description

To get a feeling for the structure of a unit description, which is pretty complex, it would be a good start to study the six elements under PrintCapabilities.

Header
This rather small section describes general attributes required for printing.

Features
This is one of the biggest sections in the schema.
Features are attributes a driver can add to a document or job without caring about the objects on the pages. There are basically three types of features.

A generic feature (exact element name: GenericFeature) is the structure, which can be used to describe any feature a printer can perform.
It is strongly recommended to name the feature compliant to the Semantic Model, if one is available there. You should also refer to settings in the Semantic Model, where applicable.

Then there is a large list of predefined features like MediaSize. Reasons not to declare a feature generic but predefined are that additional elements or attributes were considered necessary or at least helpful to better describe the feature or that we wanted to rely on a predefined name for certain features popular in drivers, as eventually the element names of features provide the connection for bi-directional communication.

A very special feature in the UPDF concept is the composite feature (exact element name: CompositeFeature).
This feature is not a new feature in the sense that it describes just another printer feature. It is rather an artificial feature used in the driver's user interface, mainly to select a number of features by one control. This is one of the most powerful tools to form the special appearance of a driver.

Objects
These are objects on the page.
The actual level of UPDF describes raster and font objects, but not vector objects.

Dependencies
Dependencies are another corner stone of the UPDF concept.
The objectives were to design them central, compact and with powerful actions typical to a driver.

There are two locations dependencies can be assigned: Either they belong to the feature they tell about or they are gathered at a central location.
Reason to go for the central solution was that a system administrator could add dependencies in a user policy. It is not allowed to re-design a feature in a user policy, but only add certain functionality like composite features or dependencies.
Additionally we think that maintenance is easier at a central location as well determining the order of dependencies.

The second objective was to decrease the number of dependencies due to related conditions. If there is a relationship between media size and duplex settings, the required actions may be slightly different depending on which feature has the actual focus. However the features involved in the special dependency are the same and the actions are supposed to be similar.
We also think that this mechanism increases the readability of dependencies.

The third objective was to provide powerful actions, as they are typical in today's drivers. You can decide on filtering out certain settings of a feature, showing messages to the end user or select specific settings of other features.

As a final note you may keep in mind that you can set dependencies for certain operating systems only, if required.

Events
As an introduction it is enough to know that events deal with the command sequences actually sent to the device. In some way you could say that you can assemble your own print file in the Events section. Typical events are the start or end of a print job or document. You will see that the list is longer.

We had an eye on optional units (e.g. an optional duplexer) in mind from beginning on. You have the chance to specify the relationship and especially position of certain events of the optional unit description and the base unit description.

UserInterfaces
This section provides the opportunity to design a driver's user interface up to some degree. It is not within the scope of UPDF to specify every single aspect of a user interface. However the description developer can group features and assign them to different levels.

You can design user interfaces for certain operating systems only, if required.

Semantic Model compliance

We spent a lot of energy in being compliant to the Semantic Model.
The UPDF schemas import the top level schema of the Semantic Model, which by itself includes further SM schemas like the ones for standardized media descriptions.

UPDF data types refer to SM data types wherever possible. Sometimes UPDF data types extend those of the Semantic Model.

UPDF element and attribute names are copied from the Semantic Model wherever possible. However they get re-defined, as we are storing additional information with them.

Please read the overview about finishing in the Common tags section for further details on common finishing structures and data types.

Free Standards Group compliance

Towards the end of the development of UPDF, version 1.0, the cooperation between the FSG and UPDF groups were intense.
We added another data type schema with additional values as used by the FSG. This schema is imported into the UPDF data types. So all values are available.

Feasibility check

To be UPDF compliant a driver has to read and understand the settings of all elements and attributes as well as the meaning of elements within the architecture.
That does not necessarily mean that a driver will be able to interpret all data entries.
Therefore it is recommended to do some kind of a feasibility check when a driver reads any type of UPDF device description.

DeviceConfiguration.Wildcards
If the setting is 'Mandatory', the driver has to be prepared to find very unique elements, which are mandatory for a proper interpretation of the data. If the driver can not interpret those tags properly, it may reject the device description as too special. That would require a dedicated driver (still UPDF compliant, of course).
This attribute does not refer to the UPDFAdditionalParameters schema with the additional parameters. These are always mandatory.
If the setting is 'Optional', the driver may ignore them, but can expect to do a reasonable job. An example could be that an IHV could add special user interface tags with more detailed specifications.

PrintCapabilities.Header.Languages
Carefully check the values here.

PhotoHalftoningRecord
Carefully check if the driver can support the specified screening matrix.

ColorRecord
Carefully check all plane related information.

RasterObject
Carefully check all settings, especially the compression.

Font handling
Device fonts are considered a special issue.
While a driver capable of handling device fonts may offer significant performance advantages, it should work correctly without them, too.

A UPDF compliant driver is expected be capable of properly interpreting all other information.

Bi-directional communication

The focus of this specification is not on bi-directional communication!

However we do consider the ID attributes of the features the identifier for a feature and the ClassifyingID attribute of the corresponding records the identifier of the feature setting. Therefore we seriously recommend to carefully make proper entries to those attributes especially when using generic features.

Every driver is supposed to know about the predefined features and their relation to the Semantic Model.

☛ The UPDF feature MediaSize corresponds to the Semantic Model feature Size.

PWG references

- "The Printer Working Group (PWG) Semantic Model 1.0 Specification ", January 20, 2004, Candidate Standard: ftp://ftp.pwg.org/pub/pwg/candidates/cs-sm10-20040120-5105.1.pdf.
-

Out of scope

As always it is difficult to decide on the right selection of features.
The following issues are considered out of scope:

- ROP codes, clipping
- Brush and other patterns

The Parameter Converter does not provide special syntax for modulo, bit shifting, absolute or negative values.

Common tags

Common attributes

You will find a number of attributes used with many elements. This section provides the common specification of these attributes. You may find further details added, where the special attribute is explained.

ID

This is supposed to be a unique identifier of a record. All ID tags are of type xsd:ID. Feel free to use numeric (not for the first character) or alphanumeric (recommended) values. While alphanumeric values are more meaningful and easier to read, you may think about some automatic conversion to more or less binary values all of the same length, if you decide not to read the XML at-run-time.
Although every developer is free to layout the architecture of the driver, it is recommended to work with the unique ID within the driver when saving or restoring settings. When communicating to the outside world in APIs to the operating system, host applications or with the device it is recommended to work with the classifying attribute ClassifyingID instead.
The ID is a mandatory attribute. One of the reasons for that design is the chance that a device description may be used by other xml files created at a later time like descriptions for optional units or user policies. In order to prepare your own description for those developers it is necessary to provide an ID with each element so that somebody can refer to it later, even if the ID is not referred to in the original description. Standard features have a fixed ID so drivers can rely on an identification.

When providing text strings for the driver's/client's user interface you refer to the ID attribute of the technical description as well. When working with the Locale instance the StringIDRef refers to an ID attribute in the technical description. The DefaultFeature and DefaultSetting attributes are references to it as well.
If an ID attribute is not resolved in a Locale instance, but the driver/client is expecting a text string, take the entry of the ID attribute in the technical description as a fallback to avoid crashes.

HelpID

The HelpID is an identifier as well as the ID and therefore a candidate to be localized.
This attribute holds the text string for the context sensitive help.

Appearance

The appearance attribute is purely used in the user interface of a driver/client.
By default most of the features are supposed to be active (Appearance = Standard), at least as long as user group policies are not in the game.
However there is the chance to specify a different appearance (Gray, Hidden, Informational) for any feature. In many cases an appearance different to 'Standard' may depend on dependencies.

This attribute specifies the appearance of the whole feature, not single records, which can be filtered by dependencies.

NonDominantRepresentative

Check the composite feature section to read about this attribute in detail. It is mandatory, except for media size and composite features. Reason: A media size is always supposed to be the dominant feature when used. A composite feature may be used driver internally, but not reported to other applications via APIs. In this case a dominant feature is not required.
Here is another sample of good habit. One may say the attribute could be optional. Agreed. But you never know whether a feature will be used as part of a composite feature in any user policy later. That's the reason why it is declared mandatory. So someone can always rely on a reasonable behavior of a feature.

ClassifyingID

The most important property of this attribute is a meaningful content. In most cases it refers to a predefined list in the UPDFDataTypes schema or PWG schemas or provides a pattern when entering other values.
This can be, but is not necessarily a unique identifier of a record.
A ClassifyingID is mandatory (except in a composite feature).

Normally each and every possible entry for a ClassifyingID has to be represented by one record. There are some exceptions.
The copy feature is the most popular. If a device is able to print 1 to 999 copies, it is not useful to create 999 records to specify that.
So the UPDF standard allows for specification of variable entries in those cases.

See the common element VariableValue for further details.

DeviceFeature

Basically every driver/client can add features that do not rely on certain device functionality.
Nonetheless some of these features are implemented in the UPDF standard, as they are commonly used.
A DeviceFeature is a mandatory attribute of boolean type, which means the value must be either ‘false’ or ‘true’.
A driver/client must check features not declared a device feature. It then can expect at least one record, which requires substitute functionality. A common sample is the N-Up feature.
If the driver/client is not able to provide that functionality, it should not support that feature nor show it in the user interface.

Common elements

VariableValue

This element is typically used when a feature with one variable value is defined. Copies are a good example.
It is comprised of the three attributes Minimum, Maximum and Step.
This element is mutually exclusive with a ClassifyingID attribute.

Simple finishing features

There are eleven finishing features, which are structured more closely to the simple way the Semantic Model is dealing with finishing. The element names all start with ‘Finishing…’, e.g. ‘FinishingStapling’. While the Semantic Model has one huge list of enumeration values for various finishing features, we have split that up into separate feature, each with a simple list of enumeration values specific to that feature. This is fine for simple finishing features like jogging, which more or less is a toggle, and for others with a simple list of values.
If the device requires a more elaborate architecture, it is recommended to go with the detailed finishing features.

Detailed finishing features

There are five finishing features, which describe specific finishing attributes in detail. The element names all start with ‘FinishingDetail…’, e.g. ‘FinishingDetailType’.
These feature can be used to describe different high level finishing features like stapling or punching. While the element names are predefined, the ID attributes are not. So one could define a FinishingDetailType with ID ‘Stapling’ and another with ID ‘Punching’.
This means you are pretty free in describing different high level finishing features.

By using a composite feature it is then possible to combine any number of finishing details to a high level finishing feature, e.g. a composite feature Stapling could well have a special FinishingDetailType as well as other details assigned.
Check out the samples on the UPDF web site.

Parameter/param:ParameterX elements

Having acknowledged the advantages of separating the real localization from the rather technical device description, it was realized that moving all PDL command sequences might have similar advantages. The PDL experts can concentrate on their area of expertise and not care about the more general device description. Additionally it would ease the duplication of a command sequence XML file for one PDL to such of another PDL.

The overall philosophy works as follows:
The initial command sequence is referenced in the Event element, as this is the place where the print file structure is composed up to a certain degree. So here you find a reference telling that a certain command sequence should appear at this position. See the event with the PDL_MediaSize ID as a sample entry. For the moment we concentrate on the CommandSequenceID of that event showing PDL_MediaSize (not necessarily the same entry as the ID).
The reference is resolved in the command sequence xml instance, where you find the same reference in a CommandSequenceID attribute of a CommandSequence element. The same element shows a CommandSequence attribute providing a more PDL like command sequence with the Parameter Converter involved.

Now we get to the part where a feature parameter is involved.
To select a specific media size it needs a variable parameter depending on the feature record selected in the driver/client. That’s why you find a term ‘MediaSize.Parameter.ID’ as part of the CommandSequence attribute entry. Now the driver/client is expecting a Parameter element with each MediaSizeRecord element. The ID entry of the selected record is finally resolved back in the CommandSequence instance to keep the real PDL stuff out of the technical description.

Assuming Letter size being selected the sample device description shows an entry ‘Param_Letter’ in the ID attribute of the Parameter element of the corresponding MediaSizeRecord. The real parameter ‘2’ is to be found in the according ResolvedParameter element of the CommandSequence instance.
This allows for optimum flexibility, while the various experts for device description or PDL support can concentrate on their business.
This is enough for the simple case.

If it becomes necessary to manage more than one parameter per feature record, you can work with the parameters in the UPDFAdditionalParameters schema (Parameter2 … Parameter9).
Please check references to Sides parameters in the CommandSequence instance.
Although the UPDFAdditionalParameters schema’s main purpose is to demonstrate how this or any other wildcard extension can be implemented, any generic UPDF driver/client is expected to understand the UPDFAdditionalParameters.xsd and therefore Parameter2 … Parameter9 elements.
See further explanations in the Device Configuration chapter.

Device Configuration Schema

The corresponding schema is named UPDFDeviceConfiguration and can found at http://www.pwg.org/schemas/upd/1.0/UPDFDeviceConfiguration.xsd.

The device configuration is the top entry point for the driver/client into UPDF.
While all other instances should be treated as read-only, this is the only file, which may be changed depending on the device's configuration.

1. DeviceConfiguration

This is the top element of the UPDF device configuration.

The element provides the following attributes:
FileTypeIdentification
UPDFVersion
UnitDescriptionReference
CommandSequencesReference (optional)
Logo (optional)
Update (optional)
Wildcards (optional)

FileTypeIdentification

The FileTypeIdentification attribute shows "Device Configuration".

UPDFVersion

The attribute tells about the UPDF version for the complete device description.

The entry identifies the minimum driver implementation of the UPDF schemas.
The current version is 1.00.00.00.
The syntax of the version value comprises of four parts.
Position1: The major version number.
Position2: The minor version number. Backward compatibility cannot be expected. Minor changes in some areas are made, but the overall concept is the same, while some structures may have changed.
Position3: Backward compatibility cannot be expected. Very small changes are made. Expected reasons are fixes of typos or similar simple mistakes. All structures are identical.
Position4: Backward compatibility can be expected. May be some enumerations have been added to a data type. Code changes are only required to support the new values.

The version of this specification will also follow this syntax. Due to editorial changes a spec may have a higher version number in the fourth part than the UPDF schema. It then replaces previous specs with the same first three parts.

A driver/client should check at least the first three parts of the version.

UnitDescriptionReference

This is the main reference and tells about the device description instance.

CommandSequencesReference

This shows the reference to the command sequence instance.

Logo

This attribute holds the name of the company logo file. Preferred files formats are jpg and tif.

Update

This attribute takes any URI, typically used to guide the user to an IHV support site.

Wildcards

Wildcard schemas are not referred to in schemas, but in instances.
The UPDF standard cannot anticipate the names of wildcard schemas. Therefore the device configuration schema does not mention wildcard schemas.
However there is the general attribute Wildcards to tell, if a driver/client must be able to interpret all those proprietary elements and attributes.

The attribute provides two settings: Mandatory and Optional (default).
With the exception of the UPDFAdditionalParameters schema, which always is considered mandatory, a device description developer may add various wildcard schemas, but cannot expect that each and every driver will understand and properly interpret those elements and attributes.
Each time you see a ‘param:ParameterX’ element listed, it could as well be replaced by any other element of any namespace added as a wildcard namespace.
This Wildcards attribute indicates, if the other wildcard schemas are mandatory for a proper device description or optional. A driver may reject a device description with unknown but mandatory wildcard schemas.

The element has the following sub-elements:
FileInfo (optional, single)
Locale (mandatory, multiple)
OptionalUnit (optional, multiple)
UserPolicy (optional, single)
ICCProfile (optional, multiple)
Driver (optional, multiple)

1.1. FileInfo

A description developer may save some information about the actual instance.

The element provides the following attributes:

FileVersion

FileVersion
This is the only mandatory attribute.
Follow the typical pattern for a version (like UPDFVersion).

The element has the following sub-elements:
Comment (optional, multiple)

1.1.1. Comment

You can set as many other version parameters as you want.

1.2. Locale

References to one or more locale instances are managed.
Keep in mind that the device description only has ID attributes. Any ID may be localized in a locale instance. If an ID is found in a locale instance, the driver should take the ID entry of the device description, which still is better than no text string (another reason to put some effort into reasonable ID entries).

The element provides the following attributes:
LocaleReference

LocaleReference
This attribute refers to one locale instance. Each locale identifies itself with a language and country attribute.

1.3. OptionalUnit

A key feature of UPDF is the idea of mirroring the real world.
While there is the basic device described in the main unit description referred to under element DeviceConfiguration, typically there are a number of optional units (duplexer, finishing unit, etc.).
It may help thinking of the base unit as the parent and of the optional units as children.
An optional unit is described by a subset of features based on the same unit description schema. This is expected to be a much smaller instance, but follows the same rules as the master.
By keeping the description of an optional unit separate, it’s quite simple to attach and especially detach them at any time. The driver/client should provide functionality for the action and update the device configuration.
Connectors are the method used to describe the relationship between the basic device and optional units.

The element provides the following attributes:
UnitDescriptionReference
CommandSequencesReference (optional)
ShippingUnit (optional)
ConnectorUnitDescriptionReference (optional)

ConnectorID (optional)

UnitDescriptionReference

This attribute refers to one mandatory unit description like for the base unit.
The entry is copied from the Option Configuration instance.

CommandSequencesReference

The optional unit may likely need additional command sequences.
The entry is copied from the Option Configuration instance.

ShippingUnit

A developer could use this attribute to indicate that this optional unit is attached to the base unit when the device is shipped. That does not automatically mean it could not be replaced later, but it could well be used for a default driver configuration.

ConnectorUnitDescriptionReference

This attribute refers to the parent unit description.
In most cases that might turn out to be the base unit description. But it could also be connected to another optional unit in a later link in the chain.

ConnectorID

This attribute refers to the ConnectorID of the parent unit description for this particular optional unit.
This and the attribute above gives the driver/client some chance to verify that a reasonable unit description has been selected.
And it tells, if a certain connector is already used or still available.

The element has the following sub-elements:
Locale (mandatory, multiple)

1.3.1. Locale

This element has the same structure as the Locale element directly under DeviceConfiguration.
In this case it refers to those locales shipped with the optional unit. These locale instances are supposed to be smaller. They do extend the basic locale instances referred to by the base unit, but do not replace those.
The entry is copied from the Option Configuration instance.

1.4. UserPolicy

User policies are normally not provided by the IHV. They are not necessary for a complete device description.
However they may turn out as a nice opportunity for operators and system administrators.
The basic idea is to allow the definition of additional dependencies, additional composite features and to replace original user interfaces of the base unit description.
Therefore a driver has to check these tags to probably change or extend the actual device description.

The element provides the following attributes:

UserPolicyDescriptionReference (optional)

UserPolicyDescriptionReference

This attribute refers to the user policy instance.

The element has the following sub-elements:
Locale (mandatory, multiple)

1.4.1. Locale

This element has the same structure as the Locale element directly under DeviceConfiguration.
In this case it refers to those locales used by the user policy.
If the user policy is referring to a locale, which is already referred to by the master description, the new locale is considered an extension.
The driver will check, if a UI string localized in the locales referred to by the master description is to be replaced due to an entry in the user policy locale. So you could replace “Letter” with “Blue Paper”.
If a UI string is not found in the master locale, it might be found in the user policy locale, especially for new composite features.

1.5. ICCProfile

You can assign multiple ICC profiles to the device configuration as well.
Enter the file name including the full path (e.g. ftp://ftp.pwg.org/pub/pwg/upd/icc1.icm), if the file is not located in the same directory as the device configuration file.

1.6. Driver

You can assign multiple additional driver components to the device configuration.
A driver component could be a special library to be installed.

The element provides the following attributes:
ID (optional)
Reference (optional)

ID

This attribute shows a driver specific identifier.
In case a description developer prefers a special UPDF capable driver, they may want that one used. To allow an installer to check a certain platform independent value, a driver should store it at the proper location (e.g. the registry under Windows).
If this attribute is not set, the installer is supposed to check, if any UPDF capable driver is installed. If not, the installer is expected to know, where to find a default UPDF driver.
It is recommended to set this attribute always.

Reference

This attribute shows the URL respectively file location, where the driver can be found.

An installer would likely check, if a workstation is connected to the Internet, if a URL is set. If not connected, it may propose a CD/DVD drive and let the end user decide on the path.

The element refers to the following sub-elements:
OperatingSystem (optional, multiple)
Component (optional, multiple)

1.6.1. OperatingSystem

This element indicates, which operating systems a special driver is determined for (e.g. MS Windows 2000/XP).

1.6.2. Component

This is the chance to list additional platform specific modules a description developer wants to have installed additionally to the driver. A future use could be the use of callbacks, where this element will likely tell about the library keeping the referred functions.
This is the file name including the full path (e.g. ftp://ftp.pwg.org/pub/pwg/upd/lib1.dll), if the file is not located in the same directory as the device configuration file.
In case a preferred driver is specified these components may come with the driver itself. This makes more sense, if no preferred driver is specified, but the description developer wants the generic UPDF driver to use an additional library.
Feel free to think of other uses.

Option Configuration Schema

The corresponding schema is named UPDFOptionConfiguration and can found at http://www.pwg.org/schemas/upd/1.0/UPDFOptionConfiguration.xsd.

2. OptionalUnitConfiguration

The element provides the following attributes:
FileTypeIdentification
UnitDescriptionReference
CommandSequencesReference (optional)

FileTypeIdentification

The FileTypeIdentification attribute shows “Option Configuration”.

UnitDescriptionReference

This attribute refers to one mandatory unit description like for the base unit.
The entry will be copied to the Device Configuration instance.

CommandSequencesReference

The optional unit may likely need additional command sequences.
The entry will be copied to the Device Configuration instance.

The element has the following sub-elements:
FileInfo (optional, single)
Locale (mandatory, multiple)
ICCProfile (optional, multiple)

2.1. FileInfo

A description developer may save some information about the actual instance.

The element provides the following attributes:
FileVersion

FileVersion

This is the only mandatory attribute.
Follow the typical pattern for a version (like UPDFVersion).

The element has the following sub-elements:
Comment (optional, multiple)

2.1.1. Comment

You can set as many other version parameters as you want.

2.2. Locale

This element has the same structure as the Locale element directly under DeviceConfiguration.
These locale instances hold the entries for the optional unit. There should be no overlapping with the locale assigned to the base unit.

2.3. ICCProfile

You can assign multiple ICC profiles to the option configuration as well.
Enter the file name including the full path (e.g. ftp://ftp.pwg.org/pub/pwg/upd/icc1.icm), if the file is not located in the same directory as the option configuration file.

Unit Description Schema

The corresponding schema is named UPDF and can found at
http://www.pwg.org/schemas/upd/1.0/UPDF.xsd.
It refers to the font handling schema named UPDFFontHandling, which can be found at
http://www.pwg.org/schemas/upd/1.0/UPDFFontHandling.xsd.
All schemas refer to the UPDFDataTypes schema, which by itself refers to the FSGDataTypes schema.
Both schemas can be found at
http://www.pwg.org/schemas/upd/1.0/UPDFDataTypes.xsd and
http://www.pwg.org/schemas/upd/1.0/FSGDataTypes.xsd respectively.

The UPDF schema is the heart of the UPDF device description.
Instances based on this schema describe the base model as well as any optional unit.

3. DeviceCapabilities

This is the top element of the UPDF device description.

The element provides the following attributes:
FileTypeIdentification

FileTypeIdentification

Each UPDF schema provides a fixed attribute for an unique file type identification. This can be used to reliably identify the schema a xml file of even a section of a xml file is based on.
In the master schema the entry is 'Master description of device or optional unit'.

The element has the following sub-elements:
FileInfo (optional, single)
DeviceHeader (mandatory, single)
PrintCapabilities (mandatory, single)
wildcards (optional, single)

3.1. FileInfo

A description developer may save some information about the actual instance.

The element provides the following attributes:
FileVersion

FileVersion

This is the only mandatory attribute.
Follow the typical pattern for a version (like UPDFVersion).

The element has the following sub-elements:
Comment (optional, multiple)

3.1.1. Comment

You can set as many other version parameters as you want.

3.2. DeviceHeader

The element provides the following attributes:
Manufacturer (optional)
ProductName (optional)
UnitName

Manufacturer

This attributes identifies the printer manufacturers name.
Sample: Company ABC, Inc.

ProductName

This attribute identifies the product name of the device.
Sample: Laser 123.

UnitName

This is a locale reference.
This attribute identifies the unit name of the device. This is currently the only mandatory field in the concept. This is the technical identifier of the device.
In case the unit is an installable option, this is the name of the option.
Sample: Laser 123 raster.

The element has the following sub-elements:
DeviceId (single, optional)
Connectors (single, optional)
DescriptionInfo (multiple, optional)

3.2.1. DeviceId

This element describes how a device is identified. This information is used as well in bi-directional communication.
For the different attributes you may check the MS Windows 98 DDK (or later versions), section ‘Printer Plug and Play Installation Requirements’.

The element provides the following attributes:
Mfc (optional)
Mdl (optional)
Cmd (optional)

Cls (optional)
Des (optional)
Cid (optional)

3.2.2. Connectors

Connectors are how optional units connect to a device or other optional units. If a device or optional unit has no child connectors defined then no optional units can be attached to it. Optional units must specify what parent connector(s) they can attach to on devices or other optional units.

A child connector may only have one optional unit connected to it at a time. If a child connector is already in use, then either the prospective optional unit must use a different child connector or the current optional unit must be uninstalled before the new optional unit may be connected.

You want to read through the documentation of the device configuration, too, and check the section on the optional units.

The element has the following sub-elements:
ConnectorForChildUnit (optional, multiple)
ConnectorToParentUnit (optional, multiple)

3.2.2.1. ConnectorForChildUnit

This element lists the connectors on this base device or optional unit that other optional units may connect to.

The element provides the following attributes:
ID

ID

See the Common Attributes section for common information about this attribute.

This is a technical identifier to specify the exact type of connector. Multiple devices and optional units may provide the same technical connector but a connector ID must be unique on a device or optional unit.

While the IHV may not necessarily know about all optional units that will be attached to the parent unit in the future, they certainly have an exact specification of the connector. As long as that is clear they don't have to know about future extensions.

A device may have a connector that is meant for one special envelope feeder. But if another company develops another optional unit that can use the exact same connector then it can easily be described and attached using UPDF.

A base device may have a child connector for an optional input tray. Optional input tray 2 may connect to this parent connector and also have the same child connector. Optional input tray 3 may also connect to this parent connector. This allows tray 2 or tray 3 to connect to the base device. If tray 2 is connected, tray 3 may connect to tray 2. If tray 3 is connected, it must be disconnected before connecting tray 2.

3.2.2.2. ConnectorToParentUnit

This element describes which connector(s) the optional unit may connect to on base devices or other optional units. Base devices do not use this element.

An optional unit may connect to multiple base devices, multiple optional units, or to a combination of base devices and optional units.

The element provides the following attributes:
ConnectorIDRef
UnitDescriptionReference

ConnectorIDRef

See the Common Attributes section for common information about this attribute.
This ID must match the ID of a ConnectorForChildUnit element in the base device or optional unit description.

UnitDescriptionReference

This attribute specifies the base device or optional unit description (UPDF) that contains the connector. If the device can connect to a connector with the same ID in multiple units then there must be multiple ConnectorToParentUnit elements. Each will specify the same ID but a different UnitDescriptionReference.

3.2.3. DescriptionInfo

This element can be used to store some information, a driver may want to show in an About box about the developer of the UPDF.
It is assumed to be used similarly to Copyright information.
This is not specified to further detail. There is no place defined to store some licence agreement information in the UPDF schemas.

This is considered technical information. It does not refer to the locale schema! It is not supposed to be localized.

3.3. PrintCapabilities

This section describes all printing capabilities.

The element has the following sub-elements:
Header (optional, single)
Features (optional, single)
Objects (optional, single)
Dependencies (optional, single)
Events (optional, single)
UserInterfaces (optional, single)

3.3.1. Header

This section contains some global printing information.

The element has the following sub-elements:
Engine (optional, single)
Languages (optional, multiple)
VirtualUnits (default 7200x7200)

3.3.1.1. Engine

The element provides the following attributes:
EngineType (optional)
EngineSpeedPPM (optional)
FacePrinted (default=Face-up)

EngineType
The attribute may indicate some general specifics of the engine.

EngineSpeedPPM
The attribute may indicate some general info about the engine speed.
Please enter a value for full simplex Letter pages (in color for a color device).

FacePrinted
The attribute indicates, whether the pages get printed Face-up or Face-down by the device. This is independent from any flipping due to an input tray.

3.3.1.1.1. Languages

This tells, which languages the device understands.
This gives the driver a chance to do some upfront checking before even starting of interpreting any detailed command sequence.

The element has the following sub-elements:
PageDescriptionLanguage (optional, single)
JobControlLanguage (optional, single)

3.3.1.1.1.1. PageDescriptionLanguage

This language mainly describes the rendering of objects on the page.

The element provides the following attributes:
PDLMajorLevel (optional)
PDLMinorLevel (optional)

PDLMajorLevel

A driver might like to know about the PDL in detail to determine proper functionality for features not described in the UPDF version 1.0 standard, e.g. download.

PDLMinorLevel

It may be useful to identify even minor level differences in the PDL. This could tell about very detailed model specific differences.

3.3.1.1.1.2. JobControlLanguage

This language mainly describes attributes, which are common for the whole job. It may be the high-level protocol for bi-directional communication as well.

The element provides the following attributes:
JCLMajorLevel (optional)
JCLMinorLevel (optional)

JCLMajorLevel

The attribute indicates the major level of a Job Control Language.

JCLMinorLevel

It may be useful to identify even minor level differences in the JCL. This could tell about very detailed model specific differences.
It is not the intention of UPDF to specify the syntax of any Job Control Language.

3.3.1.2. VirtualUnits

The virtual units are used as a reference for some attributes to specify entries independent from any device or raster resolution.
A sample is a width table for bitmap fonts. A bitmap font is described related to the resolution it is designed for. A character ‘A’ may have a width of 30 pixel in 300 dpi and a width of 60 pixel in 600 dpi. It saves time and space to specify the two values related to a more virtual resolution, e.g. 7200.
The virtual width of the character would be 7200/(300/30)=720. This only needs one entry. Now it is easy to recalculate the virtual value to a real pixel value by incorporating the current device resolution.

3.3.2. Features

The element has the following sub-elements:
RAM (optional, single)
AvailableMemory (optional, single)
PrinterResolution (optional, single)
PhotoHalftoning (optional, single)
GraphicsHalftoning (optional, single)
TextHalftoning (optional, single)
Color (optional, single)

Copies (optional, single)
Collation (optional, single)
OrientationRequested (optional, single)
MediaPageRotation (optional, single)
MediaSize (optional, single)
MediaHardwareMargins (optional, single)
MediaType (optional, single)
MediaInputTrayCheck (optional, single)
OutputBin (optional, single)
MediaFrontCoating (optional, single)
MediaBackCoating (optional, single)
MediaColor (optional, single)
MediaWeightMetric (optional, single)
Sides (optional, single)
PageOrderReceived (optional, single)
NumberUp (optional, single)
NumberUpBorder (optional, single)
PresentationDirectionNumberUp (optional, single)
MediaTargetSize (optional, single)
ScalingType (optional, single)
ScalingPercentage (optional, single)
FinishingBaling (optional, single)
FinishingBinding (optional, single)
FinishingBooklet (optional, single)
FinishingBookletBinding (optional, single)
FinishingCover (optional, single)
FinishingFolding (optional, single)
FinishingJogging (optional, single)
FinishingPunching (optional, single)
FinishingStapling (optional, single)
FinishingStitching (optional, single)
FinishingTrimming (optional, single)
FinishingDetailPosition (optional, multiple)
FinishingDetailReferenceEdge (optional, multiple)
FinishingDetailType (optional, multiple)
FinishingDetailCount (optional, multiple)
FinishingDetailAngle (optional, multiple)
GenericFeature (optional, multiple)
CompositeFeature (optional, multiple)

3.3.2.1. RAM

The element provides the following attributes:
ID
HelpID (optional)
Appearance (optional)

NonDominantRepresentative

ID

HelpID
See the Common Attributes section for common information about this attribute.

Appearance
See the Common Attributes section for common information about this attribute.

NonDominantRepresentative
See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
RAMRecord (mandatory, multiple)
wildcards (optional, single)

3.3.2.1.1. RAMRecord

The element provides the following attributes:
ID
ClassifyingID

ID
See the Common Attributes section for common information about this attribute.

ClassifyingID
See the Common Attributes section for common information about this attribute.
A pattern is provided to help the driver interpret the value.

3.3.2.1.2. wildcards

See the Common Elements section for information about these elements.

3.3.2.2. AvailableMemory

The element provides the following attributes:
ID
HelpID (optional)
Appearance (optional)
NonDominantRepresentative (optional)

ID
See the Common Attributes section for common information about this attribute.

HelpID
See the Common Attributes section for common information about this attribute.

Appearance
See the Common Attributes section for common information about this attribute.

NonDominantRepresentative
See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
AvailableMemoryRecord (mandatory, multiple)
wildcards (optional, single)

3.3.2.2.1. AvailableMemoryRecord

The element provides the following attributes:
ID

ID
See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
VariableValue

3.3.2.2.1.1. VariableValue

See the Common Elements section for common information about this element.

The element provides the following attributes:
Minimum
Maximum
Step

The attributes are explained under the VariableValue element in the Common Elements section.

3.3.2.2.2. wildcards

See the Common Elements section for information about these elements.

3.3.2.3. PrinterResolution

The element provides the following attributes:
ID
HelpID (optional)
Appearance (optional)

NonDominantRepresentative

ID
See the Common Attributes section for common information about this attribute.

HelpID
See the Common Attributes section for common information about this attribute.

Appearance
See the Common Attributes section for common information about this attribute.

NonDominantRepresentative
See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
PrinterResolutionRecord (mandatory, multiple)
wildcards (optional, single)

3.3.2.3.1. PrinterResolutionRecord

The element provides the following attributes:
ID
ClassifyingID

ID
See the Common Attributes section for common information about this attribute.

ClassifyingID
See the Common Attributes section for common information about this attribute.

The element refers to the following sub-elements:
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.2.3.1.1. Parameter/param:ParameterX

See the Common Elements section for information about these elements.

3.3.2.3.2. wildcards

See the Common Elements section for information about these elements.

3.3.2.4. PhotoHalftoning

The element provides the following attributes:

ID
HelpID (optional)
Appearance (optional)
NonDominantRepresentative

ID

See the Common Attributes section for common information about this attribute.

HelpID

See the Common Attributes section for common information about this attribute.

Appearance

See the Common Attributes section for common information about this attribute.

NonDominantRepresentative

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
PhotoHalftoningRecord (mandatory, multiple)
wildcards (optional, single)

3.3.2.4.1. PhotoHalftoningRecord

The element provides the following attributes:
ID
ClassifyingID

ID

See the Common Attributes section for common information about this attribute.

ClassifyingID

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
OneHorizontalLineOfTheHalftoningMatrix (multiple, optional) | DeviceHalftoning (mandatory, single)

3.3.2.4.1.1. OneHorizontalLineOfTheHalftoningMatrix

We are facing a XML limitation here, as we cannot define a two-dimensional array. The only chance is to define the halftoning matrix line by line. One line is a comma separated list of entries.
This element is supposed to be used, if not all of the bits per color can be resolved in the color space conversion, as the device is not capable of dealing with the required bit depth (see BitsPerPixel in the ColorRecord description).

3.3.2.4.1.2. DeviceHalftoning

The element refers to the following sub-elements:
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.2.4.1.2.1. Parameter/param:ParameterX

See the Common Elements section for information about these elements.

3.3.2.4.2. wildcards

See the Common Elements section for information about these elements.

3.3.2.5. GraphicsHalftoning

The element provides the following attributes:
ID
HelpID (optional)
Appearance (optional)
NonDominantRepresentative

ID

See the Common Attributes section for common information about this attribute.

HelpID

See the Common Attributes section for common information about this attribute.

Appearance

See the Common Attributes section for common information about this attribute.

NonDominantRepresentative

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
GraphicsHalftoningRecord (mandatory, multiple)
wildcards (optional, single)

3.3.2.5.1. GraphicsHalftoningRecord

The element provides the following attributes:
ID
ClassifyingID

ID

See the Common Attributes section for common information about this attribute.

ClassifyingID

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
DeviceHalftoning (mandatory, single)

3.3.2.5.1.1. DeviceHalftoning

The element refers to the following sub-elements:
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.2.5.1.1.1. Parameter/param:ParameterX

See the Common Elements section for information about these elements.

3.3.2.5.2. wildcards

See the Common Elements section for information about these elements.

3.3.2.6. TextHalftoning

The element provides the following attributes:
ID
HelpID (optional)
Appearance (optional)
NonDominantRepresentative

ID

See the Common Attributes section for common information about this attribute.

HelpID

See the Common Attributes section for common information about this attribute.

Appearance

See the Common Attributes section for common information about this attribute.

NonDominantRepresentative

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
TextHalftoningRecord (mandatory, multiple)
wildcards (optional, single)

3.3.2.6.1. TextHalftoningRecord

The element provides the following attributes:
ID
ClassifyingID

ID

See the Common Attributes section for common information about this attribute.

ClassifyingID

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
DeviceHalftoning (mandatory, single)

3.3.2.6.1.1. DeviceHalftoning

The element refers to the following sub-elements:
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.2.6.1.1.1. Parameter/param:ParameterX

See the Common Elements section for information about these elements.

3.3.2.6.2. wildcards

See the Common Elements section for information about these elements.

3.3.2.7. Color

The element provides the following attributes:
ID
HelpID (optional)
Appearance (optional)
NonDominantRepresentative

ID

See the Common Attributes section for common information about this attribute.

HelpID

See the Common Attributes section for common information about this attribute.

Appearance

See the Common Attributes section for common information about this attribute.

NonDominantRepresentative

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
ColorRecord (mandatory, multiple)
wildcards (optional, single)

3.3.2.7.1. ColorRecord

The element provides the following attributes:
ID
ClassifyingID
PlaneImplementation

ID

See the Common Attributes section for common information about this attribute.

ClassifyingID

See the Common Attributes section for common information about this attribute.

PlaneImplementation

The predefined data types are: PerBand, PerScanline, PerPixel.
The setting of PerPixel tells the driver there will only be one plane, which holds all information on all colors. Typical implementation is three or four byte per pixel (or one for greyscale).
The setting of PerScanline tells the driver that the device is expecting one scanline per color.
The setting of PerBand tells the driver that the device is expecting one band per color.

The element has the following sub-elements:
RGBColor (optional, multiple)
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.2.7.1.1. Parameter/param:ParameterX

See the Common Elements section for information about these elements.

3.3.2.7.1.2. RGBColor

If there are multiple RGBColor elements, the order they are specified tells the driver the order they are to be sent.

The element provides the following attributes:
Color
BitsPerPixel

Color
This is the RGB value of the device color.

BitsPerPixel
This is the bit depth of the device color.

The element refers to the following sub-elements:
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.2.7.1.2.1. Parameter/param:ParameterX
See the Common Elements section for information about these elements.

3.3.2.7.2. wildcards
See the Common Elements section for information about these elements.

3.3.2.8. Copies
This feature is clearly handled as a device feature in this level of the specification. To indicate that the device is not able to provide copies set the maximum value to 1.
This is one of two samples where the classifying ID is a positive integer (see zooming as well).

The element provides the following attributes:
ID
HelpID (optional)
Appearance (optional)
DeviceFeature (default = true)

ID
See the Common Attributes section for common information about this attribute.

HelpID
See the Common Attributes section for common information about this attribute.

Appearance
See the Common Attributes section for common information about this attribute.

DeviceFeature
See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
VariableValue (mandatory, multiple)
wildcards (optional, single)

3.3.2.8.1. VariableValue

See the Common Elements section for common information about this element.

The element provides the following attributes:
Minimum
Maximum
Step

The attributes are explained under the VariableValue element in the Common Elements section.

3.3.2.8.2. wildcards

See the Common Elements section for information about these elements.

3.3.2.9. Collation

The element provides the following attributes:
ID
HelpID (optional)
Appearance (optional)
NonDominantRepresentative
DeviceFeature (default = false)

ID

See the Common Attributes section for common information about this attribute.

HelpID

See the Common Attributes section for common information about this attribute.

Appearance

See the Common Attributes section for common information about this attribute.

NonDominantRepresentative

See the Common Attributes section for common information about this attribute.

DeviceFeature

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
CollationRecord (mandatory, multiple)
wildcards (optional, single)

3.3.2.9.1. CollationRecord

The element provides the following attributes:
ID
ClassifyingID

ID

See the Common Attributes section for common information about this attribute.

ClassifyingID

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.2.9.1.1. Parameter/param:ParameterX

See the Common Elements section for information about these elements.

3.3.2.9.2. wildcards

See the Common Elements section for information about these elements.

3.3.2.10. OrientationRequested

The element provides the following attributes:
ID
HelpID (optional)
Appearance (optional)
NonDominantRepresentative
DeviceFeature (default = true)

ID

See the Common Attributes section for common information about this attribute.

HelpID

See the Common Attributes section for common information about this attribute.

Appearance

See the Common Attributes section for common information about this attribute.

NonDominantRepresentative

See the Common Attributes section for common information about this attribute.

DeviceFeature

See the Common Attributes section for common information about this attribute.
Some devices, especially non-page printers, may only be able to provide portrait printing. A driver/client may add landscape as a non-device feature in a certain environment.

The element has the following sub-elements:
OrientationRequestedRecord (mandatory, multiple)
wildcards (optional, single)

3.3.2.10.1. OrientationRequestedRecord

The element provides the following attributes:
ID
ClassifyingID
DeviceFeature

ID

See the Common Attributes section for common information about this attribute.

ClassifyingID

See the Common Attributes section for common information about this attribute.
There are selections for portrait and landscape only. An operating system or host application is not supposed to obtain other settings.
If the device can and should handle reverse settings, this has to be handled in the MediaPageRotation feature.

The element has the following sub-elements:
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.2.10.1.1. Parameter/param:ParameterX

See the Common Elements section for information about these elements.

3.3.2.10.2. wildcards

See the Common Elements section for information about these elements.

3.3.2.11. MediaPageRotation

This feature may often be related to OrientationRequested. The two together are certainly a candidate for building a composite feature.

The element provides the following attributes:
ID
HelpID (optional)

Appearance (optional)
NonDominantRepresentative
DeviceFeature (default = false)

ID

See the Common Attributes section for common information about this attribute.

HelpID

See the Common Attributes section for common information about this attribute.

Appearance

See the Common Attributes section for common information about this attribute.

NonDominantRepresentative

See the Common Attributes section for common information about this attribute.

DeviceFeature

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
MediaPageRotationRecord (mandatory, multiple)
wildcards (optional, single)

3.3.2.11.1. MediaPageRotationRecord

The element provides the following attributes:
ID
ClassifyingID

ID

See the Common Attributes section for common information about this attribute.

ClassifyingID

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.2.11.1.1. Parameter/param:ParameterX

See the Common Elements section for information about these elements.

3.3.2.11.2. wildcards

See the Common Elements section for information about these elements.

3.3.2.12. MediaSize

Please notice that the element MediaSize corresponds to the Semantic Model element Size.

The element provides the following attributes:
ID
HelpID (optional)
Appearance (optional)
NonDominantRepresentative

ID

See the Common Attributes section for common information about this attribute.

HelpID

See the Common Attributes section for common information about this attribute.

Appearance

See the Common Attributes section for common information about this attribute.

NonDominantRepresentative

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
MediaSizeRecord (mandatory, multiple)
wildcards (optional, single)

3.3.2.12.1. MediaSizeRecord

The element provides the following attributes:
ID
ClassifyingID
FeedingOrientation (optional)
VirtualBookletSize (optional)

ID

See the Common Attributes section for common information about this attribute.

ClassifyingID

See the Common Attributes section for common information about this attribute.
‘Custom’

FeedingOrientation

Possible selections are short-edge and long-edge.

The element has the following sub-elements:
MediaHardwareMargins (0-2)
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.2.12.2. wildcards

See the Common Elements section for information about these elements.

3.3.2.13. MediaHardwareMargins

The element provides the following attributes:
ID
HelpID (optional)
Appearance (optional)

ID

See the Common Attributes section for common information about this attribute.

HelpID

See the Common Attributes section for common information about this attribute.

Appearance

See the Common Attributes section for common information about this attribute.
This feature is permanently set to hidden.
All the implied functionality is supposed to work in the background.
This feature is a top candidate to be used in dependencies to assign special margins to special sizes.

The element has the following sub-elements:
MediaHardwareMarginsRecord (mandatory, multiple)
wildcards (optional, single)

3.3.2.13.1. MediaHardwareMarginsRecord

The element provides the following attributes:
ID
ClassifyingID

ID
See the Common Attributes section for common information about this attribute.

ClassifyingID

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.2.13.1.1. Parameter/param:ParameterX

See the Common Elements section for information about these elements.

3.3.2.13.2. wildcards

See the Common Elements section for information about these elements.

3.3.2.14. MediaType

The element provides the following attributes:
ID
HelpID (optional)
Appearance (optional)
NonDominantRepresentative

ID

See the Common Attributes section for common information about this attribute.

HelpID

See the Common Attributes section for common information about this attribute.

Appearance

See the Common Attributes section for common information about this attribute.

NonDominantRepresentative

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
MediaTypeRecord (mandatory, multiple)
wildcards (optional, single)

3.3.2.14.1. MediaTypeRecord

The element provides the following attributes:
ID
ClassifyingID

ID

See the Common Attributes section for common information about this attribute.

ClassifyingID

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.2.14.1.1. Parameter/param:ParameterX

See the Common Elements section for information about these elements.

3.3.2.14.2. wildcards

See the Common Elements section for information about these elements.

3.3.2.15. MediaInputTrayCheck

The element provides the following attributes:
ID
HelpID (optional)
Appearance (optional)
NonDominantRepresentative

ID

See the Common Attributes section for common information about this attribute.

HelpID

See the Common Attributes section for common information about this attribute.

Appearance

See the Common Attributes section for common information about this attribute.

NonDominantRepresentative

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
MediaInputTrayCheckRecord (mandatory, multiple)
wildcards (optional, single)

3.3.2.15.1. MediaInputTrayCheckRecord

The element provides the following attributes:
ID
ClassifyingID

CustomSizeMinimumSize (optional)
CustomSizeMaximumSize (optional)
CustomSizeLeftOffset (optional)
CustomSizeReferenceSize (optional)
Capacity (optional)
Reservoir (optional)
MediaFlippedBeforePrinting (optional)

ID
See the Common Attributes section for common information about this attribute.

ClassifyingID
See the Common Attributes section for common information about this attribute.

CustomSizeMinimumSize
Special input trays may require special values.

CustomSizeMaximumSize
Special input trays may require special values.

Capacity
This attribute tells about the maximum capacity of the tray (standard media type).

MediaFlippedBeforePrinting
The media may get flipped once on the path from certain trays.
This is one of the paper path attributes we carry. It is recommended to fill them accurately. They may be used to indicate the proper way to insert the media into the tray or even in a larger implementation of manual duplex. See the special paragraph under the Sides feature.

The element has the following sub-elements:
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.2.15.1.1. Parameter/param:ParameterX

See the Common Elements section for information about these elements.

3.3.2.15.2. wildcards

See the Common Elements section for information about these elements.

3.3.2.16. OutputBin

The element provides the following attributes:
ID

HelpID (optional)
Appearance (optional)
NonDominantRepresentative

ID

See the Common Attributes section for common information about this attribute.

HelpID

See the Common Attributes section for common information about this attribute.

Appearance

See the Common Attributes section for common information about this attribute.

NonDominantRepresentative

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
OutputBinRecord (mandatory, multiple)
wildcards (optional, single)

3.3.2.16.1. OutputBinRecord

The element provides the following attributes:
ID
ClassifyingID
Capacity (optional)
MediaFlippedAfterPrinting (optional)

ID

See the Common Attributes section for common information about this attribute.

ClassifyingID

See the Common Attributes section for common information about this attribute.

Capacity

This attribute tells about the maximum capacity of the tray (standard media type).

MediaFlippedAfterPrinting

The media may get flipped once on the path into certain trays.
This is one of the paper path attributes we carry. It is recommended to fill them accurately. They may be used to indicate the proper way to insert the media into the tray or even in a larger implementation of manual duplex. See the special paragraph under the Sides feature.

The element has the following sub-elements:
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.2.16.1.1. Parameter/param:ParameterX

See the Common Elements section for information about these elements.

3.3.2.16.2. wildcards

See the Common Elements section for information about these elements.

3.3.2.17. MediaFrontCoating

The element provides the following attributes:
ID
HelpID (optional)
Appearance (optional)
NonDominantRepresentative

ID

See the Common Attributes section for common information about this attribute.

HelpID

See the Common Attributes section for common information about this attribute.

Appearance

See the Common Attributes section for common information about this attribute.

NonDominantRepresentative

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
MediaFrontCoatingRecord (mandatory, multiple)
wildcards (optional, single)

3.3.2.17.1. MediaFrontCoatingRecord

The element provides the following attributes:
ID
ClassifyingID

ID

See the Common Attributes section for common information about this attribute.

ClassifyingID

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.2.17.1.1. Parameter/param:ParameterX

See the Common Elements section for information about these elements.

3.3.2.17.2. wildcards

See the Common Elements section for information about these elements.

3.3.2.18. MediaBackCoating

The element provides the following attributes:
ID
HelpID (optional)
Appearance (optional)
NonDominantRepresentative

ID

See the Common Attributes section for common information about this attribute.

HelpID

See the Common Attributes section for common information about this attribute.

Appearance

See the Common Attributes section for common information about this attribute.

NonDominantRepresentative

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
MediaBackCoatingRecord (mandatory, multiple)
wildcards (optional, single)

3.3.2.18.1. MediaBackCoatingRecord

The element provides the following attributes:
ID
ClassifyingID

ID

See the Common Attributes section for common information about this attribute.

ClassifyingID
See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.2.18.1.1. Parameter/**param:ParameterX**

See the Common Elements section for information about these elements.

3.3.2.18.2. wildcards

See the Common Elements section for information about these elements.

3.3.2.19. MediaColor

The element provides the following attributes:
ID
HelpID (optional)
Appearance (optional)
NonDominantRepresentative

ID
See the Common Attributes section for common information about this attribute.

HelpID
See the Common Attributes section for common information about this attribute.

Appearance
See the Common Attributes section for common information about this attribute.

NonDominantRepresentative
See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
MediaColorRecord (mandatory, multiple)
wildcards (optional, single)

3.3.2.19.1. MediaColorRecord

The element provides the following attributes:
ID
ClassifyingID

ID

See the Common Attributes section for common information about this attribute.

ClassifyingID

See the Common Attributes section for common information about this attribute.
It is worth spending a minute to reflect on the media color data types.
UPDF is intended to be compliant to the PWG Semantic Model where possible. Therefore we refer to the MediaWellKnownValue schema with the corresponding predefined media color values as well as the extension pattern.
However we think that an even better way definition of values would include a RGB value of the media, not just a semantic identifier.
So we added another list of predefined values with RGB values included and a corresponding pattern.
There is no contradiction between the patterns, as you can check.
If a ClassifyingID shows an entry, which is not listed as a predefined enumeration and can neither be interpreted, the media is supposed to have RGB value of 'ffffff' (white).

The element has the following sub-elements:
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.2.19.1.1. Parameter/param:ParameterX

See the Common Elements section for information about these elements.

3.3.2.19.2. wildcards

See the Common Elements section for information about these elements.

3.3.2.20. MediaWeightMetric

The element provides the following attributes:
ID
HelpID (optional)
Appearance (optional)
NonDominantRepresentative

ID

See the Common Attributes section for common information about this attribute.

HelpID

See the Common Attributes section for common information about this attribute.

Appearance

See the Common Attributes section for common information about this attribute.

NonDominantRepresentative

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
MediaWeightMetricRecord (mandatory, multiple)
wildcards (optional, single)

3.3.2.20.1. MediaWeightMetricRecord

The element provides the following attributes:
ID
ClassifyingID

ID

See the Common Attributes section for common information about this attribute.

ClassifyingID

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.2.20.1.1. Parameter/param:ParameterX

See the Common Elements section for information about these elements.

3.3.2.20.2. wildcards

See the Common Elements section for information about these elements.

3.3.2.21. Sides

The element provides the following attributes:
ID
HelpID (optional)
Appearance (optional)
NonDominantRepresentative
DeviceFeature (default = false)

ID

See the Common Attributes section for common information about this attribute.

HelpID

See the Common Attributes section for common information about this attribute.

Appearance

See the Common Attributes section for common information about this attribute.

NonDominantRepresentative

See the Common Attributes section for common information about this attribute.

DeviceFeature

See the Common Attributes section for common information about this attribute.
If the feature is not declared a device feature the driver is supposed to check the HalfDocumentNotification attribute, interpret the paper path attributes involved and re-order page, if necessary.
In case of manual duplex the device description will likely show an event 'HalfDocument' (OnEnd of HalfDocument means the end of the first half, OnStart of HalfDocument means the start of the second half) and send a command sequence to inform the driver about the event. The device is then expecting some kind of user action (e.g. press a button) before it will continue printing the second path of the manual duplex.
Bi-directional feedback of any kind is not considered part of UPDF, version 1.0.
The driver has to check for the existence of the 'HalfDocument' event in case Sides is declared a non-device feature.
If the event is not used (and the feature is declared non-device) it means the driver has to take over the complete control of manual duplex by either showing proper dialogs or some other interaction with the end user.

The element has the following sub-elements:
SidesRecord (mandatory, multiple)
wildcards (optional, single)

3.3.2.21.1. SidesRecord

The element provides the following attributes:
ID
ClassifyingID
DuplexProcessing (optional)

ID

See the Common Attributes section for common information about this attribute.

ClassifyingID

See the Common Attributes section for common information about this attribute.

DuplexProcessing

The attribute tells how the media is handled on its paper path through the device. This is one of the paper path attributes we carry. It is recommended to fill them accurately. They may be used to indicate the proper way to insert the media into the tray or even in a larger implementation of manual duplex.

To gather all information necessary for a solid implementation of a manual duplex feature it is absolutely necessary to know exactly and in detail about the way the media is inserted into the input tray and the paper path it is taking, until it rests in the output bin.
Only then a functionality can be built, which is implementation independent based on the various attributes involved.
The UPDF standard provides all necessary attributes about the paper path. We do not deal with pictures and strings, which may be required to populate a proper dialog. These are supposed to be made available by the operating system, the spooler (the author's favorite) or the driver.

The element has the following sub-elements:
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.2.21.1.1. Parameter/param:ParameterX

See the Common Elements section for information about these elements.

3.3.2.21.2. wildcards

See the Common Elements section for information about these elements.

3.3.2.22. PageOrderReceived

The element provides the following attributes:
ID
HelpID (optional)
Appearance (optional)
NonDominantRepresentative
DeviceFeature (default = false)

ID

See the Common Attributes section for common information about this attribute.

HelpID

See the Common Attributes section for common information about this attribute.

Appearance

See the Common Attributes section for common information about this attribute.

NonDominantRepresentative

See the Common Attributes section for common information about this attribute.

DeviceFeature

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
PageOrderReceivedRecord (mandatory, multiple)
wildcards (optional, single)

3.3.2.22.1. PageOrderReceivedRecord

The element provides the following attributes:
ID
ClassifyingID

ID

See the Common Attributes section for common information about this attribute.

ClassifyingID

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.2.22.1.1. Parameter/param:ParameterX

See the Common Elements section for information about these elements.

3.3.2.22.2. wildcards

See the Common Elements section for information about these elements.

3.3.2.23. NumberUp

The element provides the following attributes:
ID
HelpID (optional)
Appearance (optional)
NonDominantRepresentative
DeviceFeature (default = false)

ID

See the Common Attributes section for common information about this attribute.

HelpID

See the Common Attributes section for common information about this attribute.

Appearance

See the Common Attributes section for common information about this attribute.

NonDominantRepresentative

See the Common Attributes section for common information about this attribute.

DeviceFeature

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
NumberUpRecord (mandatory, multiple)
wildcards (optional, single)

3.3.2.23.1. NumberUpRecord

The element provides the following attributes:
ID
ClassifyingID
HorizontalSpaceBetweenPages (optional)
VerticalSpaceBetweenPages (optional)
LeftOffsetFirstPage (optional)
TopOffsetFirstPage (optional)

ID

See the Common Attributes section for common information about this attribute.

ClassifyingID

See the Common Attributes section for common information about this attribute.

HorizontalSpaceBetweenPages

This attribute indicates the default space a description developer likes to see between thumb nails.
You never know, whether a driver will use or overwrite them.

VerticalSpaceBetweenPages

This attribute indicates the default space a description developer likes to see between thumb nails.
You never know, whether a driver will use or overwrite them.

LeftOffsetFirstPage

This attribute indicates the distance from the left edge of the media a description developer likes to see for thumb nails. Be careful not to specify less than the minimum hardware margin.
You never know, whether a driver will use or overwrite them.

TopOffsetFirstPage

This attribute indicates the distance from the top edge of the media a description developer likes to see for thumb nails. Be careful not to specify less than the minimum hardware margin.
You never know, whether a driver will use or overwrite them.

The element has the following sub-elements:

Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.2.23.1.1. Parameter/param:ParameterX

See the Common Elements section for information about these elements.

3.3.2.23.2. wildcards

See the Common Elements section for information about these elements.

3.3.2.24. NumberUpBorder

The element provides the following attributes:
ID
HelpID (optional)
Appearance (optional)
NonDominantRepresentative
DeviceFeature (default = false)

ID

See the Common Attributes section for common information about this attribute.

HelpID

See the Common Attributes section for common information about this attribute.

Appearance

See the Common Attributes section for common information about this attribute.

NonDominantRepresentative

See the Common Attributes section for common information about this attribute.

DeviceFeature

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
NumberUpBorderRecord (mandatory, multiple)
wildcards (optional, single)

3.3.2.24.1. NumberUpBorderRecord

The element provides the following attributes:
ID
ClassifyingID

ID

See the Common Attributes section for common information about this attribute.

ClassifyingID

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.2.24.1.1. Parameter/param:ParameterX

See the Common Elements section for information about these elements.

3.3.2.24.2. wildcards

See the Common Elements section for information about these elements.

3.3.2.25. PresentationDirectionNumberUp

The element provides the following attributes:
ID
HelpID (optional)
Appearance (optional)
NonDominantRepresentative
DeviceFeature (default = false)

ID

See the Common Attributes section for common information about this attribute.

HelpID

See the Common Attributes section for common information about this attribute.

Appearance

See the Common Attributes section for common information about this attribute.

NonDominantRepresentative

See the Common Attributes section for common information about this attribute.

DeviceFeature

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
PresentationDirectionNumberUpRecord (mandatory, multiple)
wildcards (optional, single)

3.3.2.25.1. PresentationDirectionNumberUpRecord

The element provides the following attributes:
ID
ClassifyingID

ID

See the Common Attributes section for common information about this attribute.

ClassifyingID

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.2.25.1.1. Parameter/param:ParameterX

See the Common Elements section for information about these elements.

3.3.2.25.2. wildcards

See the Common Elements section for information about these elements.

3.3.2.26. MediaTargetSize

The element provides the following attributes:
ID
HelpID (optional)
Appearance (optional)
NonDominantRepresentative

ID

See the Common Attributes section for common information about this attribute.

HelpID

See the Common Attributes section for common information about this attribute.

Appearance

See the Common Attributes section for common information about this attribute.

NonDominantRepresentative

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
MediaTargetSizeRecord (mandatory, multiple)
wildcards (optional, single)

3.3.2.26.1. MediaTargetSizeRecord

The element provides the following attributes:
ID
ClassifyingID

ID

See the Common Attributes section for common information about this attribute.

ClassifyingID

See the Common Attributes section for common information about this attribute.

3.3.2.26.2. wildcards

See the Common Elements section for information about these elements.

3.3.2.27. ScalingType

The element provides the following attributes:
ID
HelpID (optional)
Appearance (optional)
NonDominantRepresentative
DeviceFeature (default = false)

ID

See the Common Attributes section for common information about this attribute.

HelpID

See the Common Attributes section for common information about this attribute.

Appearance

See the Common Attributes section for common information about this attribute.

NonDominantRepresentative

See the Common Attributes section for common information about this attribute.

DeviceFeature

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:

ScalingTypeRecord (mandatory, multiple)
wildcards (optional, single)

3.3.2.27.1. ScalingTypeRecord

The element provides the following attributes:
ID
ClassifyingID
ScalingPosition (optional)

ID

See the Common Attributes section for common information about this attribute.

ClassifyingID

See the Common Attributes section for common information about this attribute.

ScalingPosition

The attribute tells about the position on the page after scaling.

The element has the following sub-elements:
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.2.27.1.1. Parameter/param:ParameterX

See the Common Elements section for information about these elements.

3.3.2.27.2. wildcards

See the Common Elements section for information about these elements.

3.3.2.28. ScalingPercentage

The element provides the following attributes:
ID
HelpID (optional)
Appearance (optional)
DeviceFeature (default = false)

ID

See the Common Attributes section for common information about this attribute.

HelpID

See the Common Attributes section for common information about this attribute.

Appearance

See the Common Attributes section for common information about this attribute.

DeviceFeature

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
VariableValue (mandatory, multiple)
wildcards (optional, single)

3.3.2.28.1. VariableValue

See the Common Elements section for common information about this element.

The element provides the following attributes:
Minimum
Maximum
Step

The attributes are explained under the VariableValue element in the Common Elements section.

3.3.2.28.2. wildcards

See the Common Elements section for information about these elements.

3.3.2.29. FinishingBaling

It is recommended to read the general information about finishing provided in the Common Tags section first.

The element provides the following attributes:
ID
HelpID (optional)
Appearance (optional)
NonDominantRepresentative

ID

See the Common Attributes section for common information about this attribute.

HelpID

See the Common Attributes section for common information about this attribute.

Appearance

See the Common Attributes section for common information about this attribute.

NonDominantRepresentative

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
FinishingBalingRecord (mandatory, multiple)
wildcards (optional, single)

3.3.2.29.1. FinishingBalingRecord

The element provides the following attributes:
ID
ClassifyingID

ID

See the Common Attributes section for common information about this attribute.

ClassifyingID

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.2.29.1.1. Parameter/param:ParameterX

See the Common Elements section for information about these elements.

3.3.2.29.2. wildcards

See the Common Elements section for information about these elements.

3.3.2.30. FinishingBinding

It is recommended to read the general information about finishing provided in the Common Tags section first.

The element provides the following attributes:
ID
HelpID (optional)
Appearance (optional)
NonDominantRepresentative

ID

See the Common Attributes section for common information about this attribute.

HelpID
See the Common Attributes section for common information about this attribute.

Appearance
See the Common Attributes section for common information about this attribute.

NonDominantRepresentative
See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
FinishingBindingRecord (mandatory, multiple)
wildcards (optional, single)

3.3.2.30.1. FinishingBindingRecord

The element provides the following attributes:
ID
ClassifyingID

ID
See the Common Attributes section for common information about this attribute.

ClassifyingID
See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.2.30.1.1. Parameter/param:ParameterX

See the Common Elements section for information about these elements.

3.3.2.30.2. wildcards

See the Common Elements section for information about these elements.

3.3.2.31. FinishingBooklet

It is recommended to read the general information about finishing provided in the Common Tags section first.

There are various ways to implement booklet printing.
This standard provides all components necessary to describe the functionality as required.

While the booklet feature itself is of boolean type, some developers may expect only certain media sizes to be used for booklet printing. Please define dependencies for that.

The specific implementation of booklet printing by a driver or spooler is not really important for this standard. Some may take the media size selected in the driver and announce it with a smaller scale to the applications, some may take the application objects and scale them down properly.

Depending on the implementation there might be a number of other attributes involved like a special offset from the center or some calculation of a offset, if there is more than one sheet involved in the folding. This standard does not deal with that information.

The element provides the following attributes:
ID
HelpID (optional)
Appearance (optional)
NonDominantRepresentative
DeviceFeature (default = false)

ID
See the Common Attributes section for common information about this attribute.

HelpID
See the Common Attributes section for common information about this attribute.

Appearance
See the Common Attributes section for common information about this attribute.

NonDominantRepresentative
See the Common Attributes section for common information about this attribute.

DeviceFeature
See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
FinishingBookletRecord (mandatory, multiple)
wildcards (optional, single)

3.3.2.31.1. FinishingBookletRecord

The element provides the following attributes:
ID
ClassifyingID

ID
See the Common Attributes section for common information about this attribute.

ClassifyingID

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.2.31.1.1. Parameter/param:ParameterX

See the Common Elements section for information about these elements.

3.3.2.31.2. wildcards

See the Common Elements section for information about these elements.

3.3.2.32. FinishingBookletBinding

It is recommended to read the general information about finishing provided in the Common Tags section first.

The booklet binding feature generally tells about the start page.

The element provides the following attributes:
ID
HelpID (optional)
Appearance (optional)
NonDominantRepresentative
DeviceFeature (default = false)

ID

See the Common Attributes section for common information about this attribute.

HelpID

See the Common Attributes section for common information about this attribute.

Appearance

See the Common Attributes section for common information about this attribute.

NonDominantRepresentative

See the Common Attributes section for common information about this attribute.

DeviceFeature

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:

FinishingBookletBindingRecord (mandatory, multiple)
wildcards (optional, single)

3.3.2.32.1. FinishingBookletBindingRecord

The element provides the following attributes:
ID
ClassifyingID

ID

See the Common Attributes section for common information about this attribute.

ClassifyingID

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.2.32.1.1. Parameter/param:ParameterX

See the Common Elements section for information about these elements.

3.3.2.32.2. wildcards

See the Common Elements section for information about these elements.

3.3.2.33. FinishingCover

It is recommended to read the general information about finishing provided in the Common Tags section first.

This feature is treated as a boolean.
If the driver or spooler likes to provide extra functionality to select the media from a special tray or even select certain information to be placed on the cover, this standard is not dealing with that kind of information.
However a tray selection can be realized up to some degree with a proper use of a composite feature.

The element provides the following attributes:
ID
HelpID (optional)
Appearance (optional)
NonDominantRepresentative

ID

See the Common Attributes section for common information about this attribute.

HelpID

See the Common Attributes section for common information about this attribute.

Appearance

See the Common Attributes section for common information about this attribute.

NonDominantRepresentative

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
FinishingCoverRecord (mandatory, multiple)
wildcards (optional, single)

3.3.2.33.1. FinishingCoverRecord

The element provides the following attributes:
ID
ClassifyingID

ID

See the Common Attributes section for common information about this attribute.

ClassifyingID

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.2.33.1.1. Parameter/param:ParameterX

See the Common Elements section for information about these elements.

3.3.2.33.2. wildcards

See the Common Elements section for information about these elements.

3.3.2.34. FinishingFolding

It is recommended to read the general information about finishing provided in the Common Tags section first.

The element provides the following attributes:

ID
HelpID (optional)
Appearance (optional)
NonDominantRepresentative

ID
See the Common Attributes section for common information about this attribute.

HelpID
See the Common Attributes section for common information about this attribute.

Appearance
See the Common Attributes section for common information about this attribute.

NonDominantRepresentative
See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
FinishingFoldingRecord (mandatory, multiple)
wildcards (optional, single)

3.3.2.34.1. FinishingFoldingRecord

The element provides the following attributes:
ID
ClassifyingID

ID
See the Common Attributes section for common information about this attribute.

ClassifyingID
See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.2.34.1.1. Parameter/param:ParameterX

See the Common Elements section for information about these elements.

3.3.2.34.2. wildcards

See the Common Elements section for information about these elements.

3.3.2.35. FinishingJogging

It is recommended to read the general information about finishing provided in the Common Tags section first.

The element provides the following attributes:
ID
HelpID (optional)
Appearance (optional)
NonDominantRepresentative

ID

See the Common Attributes section for common information about this attribute.

HelpID

See the Common Attributes section for common information about this attribute.

Appearance

See the Common Attributes section for common information about this attribute.

NonDominantRepresentative

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
FinishingJoggingRecord (mandatory, multiple)
wildcards (optional, single)

3.3.2.35.1. FinishingJoggingRecord

The element provides the following attributes:
ID
ClassifyingID

ID

See the Common Attributes section for common information about this attribute.

ClassifyingID

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.2.35.1.1. Parameter/param:ParameterX

See the Common Elements section for information about these elements.

3.3.2.35.2. wildcards

See the Common Elements section for information about these elements.

3.3.2.36. FinishingPunching

It is recommended to read the general information about finishing provided in the Common Tags section first.

The element provides the following attributes:
ID
HelpID (optional)
Appearance (optional)
NonDominantRepresentative

ID

See the Common Attributes section for common information about this attribute.

HelpID

See the Common Attributes section for common information about this attribute.

Appearance

See the Common Attributes section for common information about this attribute.

NonDominantRepresentative

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
FinishingPunchingRecord (mandatory, multiple)
wildcards (optional, single)

3.3.2.36.1. FinishingPunchingRecord

The element provides the following attributes:
ID
ClassifyingID

ID

See the Common Attributes section for common information about this attribute.

ClassifyingID

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:

Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.2.36.1.1. Parameter/param:ParameterX

See the Common Elements section for information about these elements.

3.3.2.36.2. wildcards

See the Common Elements section for information about these elements.

3.3.2.37. FinishingStapling

It is recommended to read the general information about finishing provided in the Common Tags section first.

The element provides the following attributes:
ID
HelpID (optional)
Appearance (optional)
NonDominantRepresentative

ID

See the Common Attributes section for common information about this attribute.

HelpID

See the Common Attributes section for common information about this attribute.

Appearance

See the Common Attributes section for common information about this attribute.

NonDominantRepresentative

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
FinishingStaplingRecord (mandatory, multiple)
wildcards (optional, single)

3.3.2.37.1. FinishingStaplingRecord

The element provides the following attributes:
ID
ClassifyingID

ID

See the Common Attributes section for common information about this attribute.

ClassifyingID
See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.2.37.1.1. Parameter/param:ParameterX

See the Common Elements section for information about these elements.

3.3.2.37.2. wildcards

See the Common Elements section for information about these elements.

3.3.2.38. FinishingStitching

It is recommended to read the general information about finishing provided in the Common Tags section first.

The element provides the following attributes:
ID
HelpID (optional)
Appearance (optional)
NonDominantRepresentative

ID
See the Common Attributes section for common information about this attribute.

HelpID
See the Common Attributes section for common information about this attribute.

Appearance
See the Common Attributes section for common information about this attribute.

NonDominantRepresentative
See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
FinishingStitchingRecord (mandatory, multiple)
wildcards (optional, single)

3.3.2.38.1. FinishingStitchingRecord

The element provides the following attributes:
ID
ClassifyingID

ID

See the Common Attributes section for common information about this attribute.

ClassifyingID

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.2.38.1.1. Parameter/param:ParameterX

See the Common Elements section for information about these elements.

3.3.2.38.2. wildcards

See the Common Elements section for information about these elements.

3.3.2.39. FinishingTrimming

It is recommended to read the general information about finishing provided in the Common Tags section first.

The element provides the following attributes:
ID
HelpID (optional)
Appearance (optional)
NonDominantRepresentative

ID

See the Common Attributes section for common information about this attribute.

HelpID

See the Common Attributes section for common information about this attribute.

Appearance

See the Common Attributes section for common information about this attribute.

NonDominantRepresentative

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
FinishingTrimmingRecord (mandatory, multiple)
wildcards (optional, single)

3.3.2.39.1. FinishingTrimmingRecord

The element provides the following attributes:
ID
ClassifyingID

ID

See the Common Attributes section for common information about this attribute.

ClassifyingID

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.2.39.1.1. Parameter/param:ParameterX

See the Common Elements section for information about these elements.

3.3.2.39.2. wildcards

See the Common Elements section for information about these elements.

3.3.2.40. FinishingDetailPosition

It is recommended to read the general information about finishing provided in the Common Tags section first.

The feature tells about the first finishing part on the media corresponding to the reference edge specified in FinishingDetailReferenceEdge.
Sample: A value of '10.000mm' with a top reference edge indicates the position of the first finishing part, e.g. a staple. It does not tell at all about other staples nor about the distance from the left edge of the media in case the FinishingDetailType is set to 'Side'.

The element provides the following attributes:
ID
HelpID (optional)
Appearance (optional)
NonDominantRepresentative

ID

See the Common Attributes section for common information about this attribute.
As with every other finishing feature providing info on finishing details this attribute does not have a fixed entry. It is possible to define one element for a high level finishing feature punching (and give it an ID of FDP_Punching) and another for stapling (and give it an ID of FDP_Stapling). And so on.

HelpID

See the Common Attributes section for common information about this attribute.

Appearance

See the Common Attributes section for common information about this attribute.
In case these feature are designed to tell about the way a specific high level finishing feature works as a part of a composite feature, this element may be set to 'Hidden'.

NonDominantRepresentative

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
FinishingDetailPositionRecord (mandatory, multiple)
wildcards (optional, single)

3.3.2.40.1. FinishingDetailPositionRecord

The element provides the following attributes:
ID
ClassifyingID

ID

See the Common Attributes section for common information about this attribute.

ClassifyingID

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.2.40.1.1. Parameter/param:ParameterX

See the Common Elements section for information about these elements.

3.3.2.40.2. wildcards

See the Common Elements section for information about these elements.

3.3.2.41. FinishingDetailReferenceEdge

It is recommended to read the general information about finishing provided in the Common Tags section first.

This is the reference edge for the position of the first finishing part. More details under FinishingDetailPosition.

The element provides the following attributes:
ID
HelpID (optional)
Appearance (optional)
NonDominantRepresentative

ID

See the Common Attributes section for common information about this attribute.
As with every other finishing feature providing info on finishing details this attribute does not have a fixed entry. It is possible to define one element for a high level finishing feature punching (and give it an ID of FDR_Punching) and another for stapling (and give it an ID of FDR_Stapling). And so on.

HelpID

See the Common Attributes section for common information about this attribute.

Appearance

See the Common Attributes section for common information about this attribute.
In case these feature are designed to tell about the way a specific high level finishing feature works as a part of a composite feature, this element may be set to ‘Hidden’.

NonDominantRepresentative

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
FinishingDetailReferenceEdgeRecord (mandatory, multiple)
wildcards (optional, single)

3.3.2.41.1. FinishingDetailReferenceEdgeRecord

The element provides the following attributes:
ID
ClassifyingID

ID

See the Common Attributes section for common information about this attribute.

ClassifyingID

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.2.41.1.1. Parameter/param:ParameterX

See the Common Elements section for information about these elements.

3.3.2.41.2. wildcards

See the Common Elements section for information about these elements.

3.3.2.42. FinishingDetailType

It is recommended to read the general information about finishing provided in the Common Tags section first.

This feature describes the general layout of the finishing detail. Typical settings are 'Side' and 'Corner'.

The element provides the following attributes:
ID
HelpID (optional)
Appearance (optional)
NonDominantRepresentative

ID

See the Common Attributes section for common information about this attribute.
As with every other finishing feature providing info on finishing details this attribute does not have a fixed entry. It is possible to define one element for a high level finishing feature punching (and give it an ID of FDT_Punching) and another for stapling (and give it an ID of FDT_Stapling). And so on.

HelpID

See the Common Attributes section for common information about this attribute.

Appearance

See the Common Attributes section for common information about this attribute.
In case these feature are designed to tell about the way a specific high level finishing feature works as a part of a composite feature, this element may be set to 'Hidden'.

NonDominantRepresentative

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
FinishingDetailTypeRecord (mandatory, multiple)
wildcards (optional, single)

3.3.2.42.1. FinishingDetailTypeRecord

The element provides the following attributes:
ID
ClassifyingID

ID

See the Common Attributes section for common information about this attribute.

ClassifyingID

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.2.42.1.1. Parameter/param:ParameterX

See the Common Elements section for information about these elements.

3.3.2.42.2. wildcards

See the Common Elements section for information about these elements.

3.3.2.43. FinishingDetailCount

It is recommended to read the general information about finishing provided in the Common Tags section first.

This feature can indicate the number of staples or punch holes.

The element provides the following attributes:
ID
HelpID (optional)
Appearance (optional)
NonDominantRepresentative

ID

See the Common Attributes section for common information about this attribute.

As with every other finishing feature providing info on finishing details this attribute does not have a fixed entry. It is possible to define one element for a high level finishing feature punching (and give it an ID of FDC_Punching) and another for stapling (and give it an ID of FDC_Stapling). And so on.

HelpID
See the Common Attributes section for common information about this attribute.

Appearance
See the Common Attributes section for common information about this attribute.
In case these feature are designed to tell about the way a specific high level finishing feature works as a part of a composite feature, this element may be set to 'Hidden'.

NonDominantRepresentative
See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
FinishingDetailCountRecord (mandatory, multiple)
wildcards (optional, single)

3.3.2.43.1. FinishingDetailCountRecord

The element provides the following attributes:
ID
ClassifyingID

ID
See the Common Attributes section for common information about this attribute.

ClassifyingID
See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.2.43.1.1. Parameter/param:ParameterX

See the Common Elements section for information about these elements.

3.3.2.43.2. wildcards

See the Common Elements section for information about these elements.

3.3.2.44. FinishingDetailAngle

It is recommended to read the general information about finishing provided in the Common Tags section first.

If used this feature indicates the angle of the finishing parts, e.g. the staples. A value of zero indicates a vertical staple. Angles are counted clockwise. Only full degrees are valid.

The element provides the following attributes:
ID
HelpID (optional)
Appearance (optional)
NonDominantRepresentative

ID

See the Common Attributes section for common information about this attribute.
As with every other finishing feature providing info on finishing details this attribute does not have a fixed entry. It is possible to define one element for a high level finishing feature punching (and give it an ID of FDA_Punching) and another for stapling (and give it an ID of FDA_Stapling). And so on.

HelpID

See the Common Attributes section for common information about this attribute.

Appearance

See the Common Attributes section for common information about this attribute.
In case these feature are designed to tell about the way a specific high level finishing feature works as a part of a composite feature, this element may be set to 'Hidden'.

NonDominantRepresentative

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
FinishingDetailAngleRecord (mandatory, multiple)
wildcards (optional, single)

3.3.2.44.1. FinishingDetailAngleRecord

The element provides the following attributes:
ID
ClassifyingID

ID

See the Common Attributes section for common information about this attribute.

ClassifyingID

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.2.44.1.1. Parameter/param:ParameterX

See the Common Elements section for information about these elements.

3.3.2.44.2. wildcards

See the Common Elements section for information about these elements.

3.3.2.45. GenericFeature

Generic features are the chance to add a new feature to the system.

The element provides the following attributes:
ID
HelpID (optional)
Appearance (optional)
NonDominantRepresentative

ID

See the Common Attributes section for common information about this attribute.

HelpID

See the Common Attributes section for common information about this attribute.

Appearance

See the Common Attributes section for common information about this attribute.

NonDominantRepresentative

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
GenericFeatureRecord (mandatory, multiple) | VariableValue (mandatory, single)
wildcards (optional, single)

3.3.2.45.1. GenericFeatureRecord

The element provides the following attributes:
ID
ClassifyingID (optional)

ID
See the Common Attributes section for common information about this attribute.

ClassifyingID

The element has the following sub-elements:
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.2.45.1.1. Parameter/param:ParameterX
See the Common Elements section for information about these elements.

3.3.2.45.2. VariableValue
See the Common Elements section for common information about this element.

The element provides the following attributes:
Minimum
Maximum
Step

The attributes are explained under the VariableValue element in the Common Elements section.

3.3.2.45.3. wildcards
See the Common Elements section for information about these elements.

3.3.2.46. CompositeFeature
You don't have to use them, but they may come in handy.
What can you do with them?
UPDF is a rather technical device description. The specified features are very much based on historic experience with Page Description Languages and Job Control Languages. This is not necessarily the set of features a driver developer wants to show in a user interface.
The basic idea of composite features is to be able to assemble any number of features to a new feature - a composite feature (sometimes abbreviated c.f. in this specification).

In this section we may refer to two typical samples of a possible implementation: a Media (with components media size, source and type) and a Print Quality (with component device resolution and color mode) composite feature. The corresponding device description would therefore show two CompositeFeature elements. You can assemble as many composite features as you like.

The element provides the following attributes:
ID

HelpID (optional)
Appearance (optional)
NonDominantRepresentative (optional)
DominantFeature (optional)
UserExtensible (optional)

ID

See the Common Attributes section for common information about this attribute.

HelpID

See the Common Attributes section for common information about this attribute.

Appearance

See the Common Attributes section for common information about this attribute.

NonDominantRepresentative

This attribute only becomes important when a feature is used as a component of a composite feature. Max one of the components can be declared dominant. If there is a dominant feature, then all the others must be non-dominant.

In our Media sample the media size is a component and declared dominant. The setting has to be communicated to the operating system and the intention of the sample that the media size control in application shall be the main selector of print media in the device.

The media source is a component as well and has to be communicated to the operating system, too. This is the challenge.

In our sample the source is already selected as a hidden attribute of the size. So we don't want anybody to mess it up by selecting something else in the application's media source control. However a driver is typically supposed to announce media sources. So we need a dummy entry, which could show the text "Selected by Size" or "AutoSelect". This entry must be created as a media source record! It corresponds to the general understanding of AutoSelect. If you already have an AutoSelect record, as your PDL is requiring that, you are fine to use that one.

For most features this attribute is mandatory, except for media size, as this special feature is supposed to be always used as the dominant feature. All features, which are candidates to being used as a non-dominant component of a composite feature, should have a dedicated dummy entry. Refer to the record, which makes the most sense of being used as an AutoSelect, in attribute NonDominantRepresentative for all other features. The attribute is mandatory, as you never know, if there will be somebody in future using this feature in a composite feature, perhaps in a user policy. So it is considered good manners to have a reference.

The attribute is optional for composite features. However it is recommended to use it and set it. The reason not to do it could be that one composite feature could be considered at the top level for eternity and would never be used by another composite feature as a component. But having user policies in mind this is courageous to say.

DominantFeature

This is a reference to one of the component features.

As long as all components of a composite feature as well as the composite feature itself are managed privately by the driver it may not be apparent that in some cases one component should somehow work as

the public representative for the composite feature. Only one component can be dominant. Makes sense, as otherwise you may select the same thing at a different place again with a different selection.
If there is a dominant feature, the driver should not show any component of the composite feature on the same selection level, but in a subdialog to either show the assembly of the composite feature for informational purpose or to allow the assembly of new records or other management tasks.

In the sample of a Media feature the dominant feature would certainly be the media size. This is likely the only sample where the dominant feature can almost be predicted. In most other implementations that could be a very arbitrary decision.
Under MS Windows a driver has to announce certain features to the operating system in a proper structure (devmode). The media size setting is one entry in that structure.
With the media size as the dominant feature in a Media c.f. the Media records would replace the original media size records.
The end user would see the Media strings like "Letter from Tray 2", but the driver would announce the classifying ID of the letter record of media size. Further elucidation under User Policies.

If there is no dominant feature declared the composition is supposed to work as a kind of a default selection for a set of features, in which case the single features may (depends on the driver) still appear on the same selection level as the composite features. It also depends on the driver's implementation, if these default values can be changed. The author of this specification would perhaps check the UserExtensible attribute to make that decision.

UserExtensible

This could turn out as an important attribute for the end user. If you read about composite features first, you may want to jump over this paragraph, study the other elements and attributes involved and then come back here. It is important that you have a good overview of c.f. to understand the power of this attribute.

The IHV is normally supposed to be the only instance to
An IHV device description developer decides on this attribute first.
If set to False, the developer has decided that the user can only use the predefined records.
This may be realistic for a composite feature for print quality.

Watch that the attribute is optional. If not set (true is considered default), the system administrator is next to decide on it in a user policy. This gives him/her some power.
If set to True, the user can assemble user defined sets and delete those (not the original ones) again if requested.
If not set at all, this is to be interpreted as True.
This is a major feature, what makes the UPDF design so powerful. A driver can be developed pretty much generically for composite features and depending on the setting of the UserExtensible attribute the user can add records.

The element has the following sub-elements:
CompositeFeatureRecord (mandatory, multiple)
wildcards (optional, single)

3.3.2.46.1. CompositeFeatureRecord

Here a record is one assembly of one special setting per components.
In our Media sample these records may be represented by a user interface string like "Letter from Tray 2", "A4 Booklet", "Ledger Short-Edge" or just "Letter".
See the explanations of the UserExtensible attribute to find out about the power a system adminstrator has.

The element provides the following attributes:
ID
ClassifyingID (optional)

ID

See the Common Attributes section for common information about this attribute.

ClassifyingID

The element has the following sub-elements:
Component (mandatory, multiple)
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.2.46.1.1. Component

Typically two or more components with special settings define a record.
A Media feature could be defined by components size, type, source and even more.
It is seriously recommended to define the same number of components per record, as the driver might rely on finding a setting per component per record.

The element provides the following attributes:
FeatureID
FeatureSetting

FeatureID

This is the chance to select a feature ID and the required setting for the component.
As a sample you may want to select the media size feature and the setting of 'Letter' for this CompositeFeatureRecord. You do this by referencing the proper feature ID of media size and the ID of the Letter record.

FeatureSetting

Some Q&A to elucidate the design.
Question: You want to re-define a feature of the main device description? Like hiding some records consistently (no dependencies involved).
Answer: Build a composite feature with that one predefined feature as the only component and do it there.

☛ You may want to make that component dominant.

As you may remember, you cannot re-define the feature itself. The records there are read-only. So this does the trick.
Of course, you could still define some dependencies instead.

Question: You want to merge the media types into the list of media sources and announce the merged list as media sources to the operating system?
Answer: A composite feature is the solution.

Question: Who does define the CompositeFeatureRecord elements?
Answer: The UPDF device description developer and may be later the user policy developer.
Once at least one record for Media is predefined - and that should be the safe default Media record - a smart driver will check, if a user policy is involved.
How? Before filling the platform structures like a devmode under MS Windows it would check, if the media size feature is used as a component. If somebody is using media size as a component in two or more composite features, the description is clearly ruined! This may not lead to a bug or crash, but to unpredictable results. If media size is used as a component, it's supposed to be the dominant, but the driver would check generally. This has to be a recursive method, as a composite feature can be used within a composite feature. Sounds complicated, but feels natural once implemented. The same is true, if other dominant features are re-used as a component on a higher level again.

☛ Statement: A composite feature can only have max one dominant feature. If this c.f. is used in another c.f., the dominant feature reaches through. There can never be a second dominant feature.

Question: Is the user stuck with the records predefined by the device description developer?
Answer: The first statement is that feature records defined in the original device description can never be deleted. However the driver may decide on offering functionality to deactivate them (implementation dependent). Another emergency exit are dependencies. But for the worst case you need at least one record under every condition as the fallback.
The user can define new records of composite features, if the UserExtensible attribute allow for it. Please read the corresponding section.

Dreams?
It would be marvellous, if the operating system would provide functionality to make the underlying attributes of a composite feature visible. That would mean somebody could do something like a right-mouse-click or corresponding functionality and then could be informed that the "Yellow Letterhead" has attributes 'size=Letter', 'source=tray2', 'media=cardstock' or something like that. The point is that the driver can fill that structure or even better is asked at run-time to do so.
It must be allowed to dream. As a matter of fact we do not think that something like composite features is at all special or limited to printer drivers.

3.3.2.46.1.2. Parameter/param:ParameterX

See the Common Elements section for information about these elements.
This element is supposed to be used rarely.
But it is possible to work with a special command sequence and according parameters for a composite feature.

Quintessence
Composite features, especially in combination with user policies are considered one of the greatest accomplishments of the UPDF standard.

3.3.2.46.2. wildcards

See the Common Elements section for information about these elements.

3.3.3. Objects

The section Objects refers to objects on the page to be printed.
We included a description about the way the device expects positioning to be sent.
The UPDF standard level 1 does not cover vector objects.

Different to features objects are not supposed to be presented in dialogs.

The element has the following sub-elements:
Positioning (optional, single)
RasterObjects (mandatory, single)
FontObjects (optional, single)
wildcards (optional, single)

3.3.3.1. Positioning

The element provides the following attributes:
ID

ID

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
PositioningRecord (mandatory, multiple)

3.3.3.1.1. PositioningRecord

The element provides the following attributes:
ID
ClassifyingID
MaximumValue
ReferenceXDimension
ReferenceYDimension
Direction

ID

See the Common Attributes section for common information about this attribute.

ClassifyingID

See the Common Attributes section for common information about this attribute.
This attribute tells about the resolution for both dimensions.

MaximumValue

This attribute tells about the maximum values for both dimensions and follows the same pattern as the ClassifyingID.

ReferenceXDimension

This attribute tells whether the device is expecting command sequences for absolute or relative positioning in the horizontal dimension.

ReferenceYDimension

This attribute tells whether the device is expecting command sequences for absolute or relative positioning in the vertical dimension.

Direction

This attribute tells, which direction the printer can position: forward (= Positive), backwards (= Negative) or both ways.

The element has the following sub-elements:
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.3.1.1.1. Parameter/param:ParameterX

See the Common Elements section for information about these elements.

3.3.3.2. RasterObjects

This is the section about raster or bitmap objects.

The element provides the following attributes:
ID

ID

See the Common Attributes section for common information about this attribute.

The element has the following sub-elements:
RasterObjectRecord (mandatory, multiple)
wildcards (optional, single)

3.3.3.2.1. RasterObjectRecord

The element provides the following attributes:
ID
ClassifyingID
ByteOrientation
CAP

ID

See the Common Attributes section for common information about this attribute.

ClassifyingID

See the Common Attributes section for common information about this attribute.

ByteOrientation

When a band contains more than one scanline the bytes may be oriented horizontally or vertically.

CAP

This attribute tells the driver where the device's cursor is at the end of the actual raster object. So the proper positioning for the next object can be determined.

The element has the following sub-elements:
Compression (optional, multiple)
Padding (optional, single)
Band (optional, single)
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.3.2.1.1. Compression

The element provides the following attributes:
ID
ClassifyingID

ID

See the Common Attributes section for common information about this attribute.

ClassifyingID

See the Common Attributes section for common information about this attribute.
It is most seriously recommended to check, if all entries are supported, before attaching a UPDF device description to a generic UPDF driver.

The element has the following sub-elements:
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.3.2.1.1.1. Parameter/param:ParameterX

See the Common Elements section for information about these elements.

3.3.3.2.1.2. Padding

The element provides the following attributes:
LeftMargin (optional)
RightMargin (optional)
ByteAlignment (optional)

LeftMargin

This boolean attribute indicates, whether the left page margin should be realized as part of the image.

RightMargin

This boolean attribute indicates, whether the left page margin should be realized as part of the image.

ByteAlignment

This attribute indicates, if there the last bits of a scanline have to be filled up to a certain value, e.g. 8.

3.3.3.2.1.3. Band

The element provides the following attributes:
Units
Size

Units

The two values possible are kilobytes of RAM or number of scanlines.

Size

This attribute determines the exact band size in the units declared in the attribute above.
If there is no Band element defined, the device accepts any band size up to the full page.

3.3.3.2.1.4. Parameter/param:ParameterX

See the Common Elements section for information about these elements.

3.3.3.2.2. wildcards

See the Common Elements section for information about these elements.
This could connect to some additional raster graphic related info.

3.3.3.3. FontObjects

UPDF supports device fonts. We specify all metrics information an operating system may be interested in. We do not store information about the shape of a glyph. Nor do we store information about the detailed font download method a PDL may use.

Some basics

There are different strategies to support device fonts.
Some drivers announce them actively.
On a platform supporting device fonts that means the user would see these fonts listed, even after having removed all system fonts.
Some other drivers like to hide device fonts from the public and use them only in case a system font with the same name and basic characteristics is used. Then the system font is substituted by the proper device font. In this specification we call these fonts passive fonts. The user would not see any of them after having deleted all system fonts.

We want to provide a modular font structure for a number of reasons.
Maintenance of the font data will be easier when the information is stored once only and not several times at different places.
So we want to support people who have built UPDF font information, let's say for an older Arial (without the Euro sign) and now do not want to save the complete information for another Arial (now including the Euro sign) again. Other scenarios possible.
As every device font has a font ID, it can be referred to by another font.
In this specification we call a font, which is referred to by another font, a subfont.
The referring font uses the FontReference element to do that.
Although a driver may only check some entries in a subfont, it is considered good behavior to fill out all entries carefully, as it could be copied somewhere else and be used as a standalone font there. The attributes not checked by a driver are supposed to be a copy of the referring font. The ruling font eventually controls all final parameters announced publically.

For understanding passive and subfonts a driver must be prepared to find a list of device fonts, where not all of them are to be announced to the operating system.
To find out which ones are the right ones the driver has to check the name attributes in the NameIDs element.

Font units

In a number of fields a numeric value is expected, which tells about font metrics. These values are supposed to be in design units for scalable fonts and in virtual units for bitmap fonts.
Sample: GlobalMetrics.MaximumHeight = 1200
 This would be the value for a 12 point high bitmap font,
assuming virtual units of 7200.
This syntax describes bitmap fonts practically independent of resolutions.
If certain or all fonts shall not be listed for certain device resolutions, this is considered a dependency and has to be handled in the dependencies section.

Modifying attributes versus identifying attributes

Identifying attributes define the unique attributes of every single font.
Modifying attributes can change every font in the same way without changing the cell dimensions.
However each font has a PseudoDeviceFonts elements telling which of the modifying attributes it accepts.

The FontObjects element refers to the following sub-elements:

ModifyingAttributes (mandatory, single)
DeviceFont (optional, multiple)
wildcards (optional, single)

3.3.3.3.1. ModifyingAttributes

There are identifying attributes of fonts as well as modifying attributes.
While the parameters of identifying attributes are unique to any device font, modifying attributes can be used the same way with all or at least with a certain group of device fonts and even download methods.
As a sample a command sequence for switching an underline feature on and off can be used by a font by assigning the corresponding On/Off parameters to the font. It is transforming the selected font, but does not change the selection at all.
It depends on the operating system, what the driver is going to do with the information about modifying attributes.
Warning: If an attribute changes the cell dimensions, it is not a modifying attribute. In this case a new device font must be defined.
Right now the modifying attributes are placed on a very high level in the font structure. The opposite extreme would be to assign it to each device font and download method separately. We wanted to save that space and do it once only.
Modifying attributes referred to by a specific font typically are a toggle to switch it on and off. A driver must clearly identify the record of such an attribute to be able to handle it correctly.

The element refers to the following sub-elements:
Slant (optional, multiple)
Weight (optional, multiple)
Underline (optional, multiple)
Strikeout (optional, multiple)
Rotation (mandatory, single)
DeviceColor (mandatory, multiple)
GlobalTextCapabilities (mandatory, single)

3.3.3.3.1.1. Slant

This section tells how to make a font artifically italic, also known as oblique, without changing the width of the characters.

The element provides the following attributes:
Slant (mandatory)

Slant

If used typically two records to switch it on and off.

The element has the following sub-elements:
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.3.3.1.1.1. Parameter/param:ParameterX

See the Common Elements section for information about these elements.

3.3.3.3.1.2. Weight

This section tells how to make a font artifically bold without changing the width of the characters.

The element provides the following attributes:
Weight (mandatory)

Weight
If used typically two records to switch it on and off.

The element has the following sub-elements:
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.3.3.1.2.1. Parameter/param:ParameterX

See the Common Elements section for information about these elements.

3.3.3.3.1.3. Underline

This section tells how to underline a font.

The element provides the following attributes:
Underline (mandatory)

Underline
If used typically two records to switch it on and off.

The element has the following sub-elements:
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.3.3.1.3.1. Parameter/param:ParameterX

See the Common Elements section for information about these elements.

3.3.3.3.1.4. Strikeout

This section tells how to strikeout a font.

The element provides the following attributes:
Strikeout (mandatory)

Strikeout

If used typically two records to switch it on and off.

The element has the following sub-elements:
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.3.3.1.4.1. Parameter/param:ParameterX

See the Common Elements section for information about these elements.

3.3.3.3.1.5. Rotation

This section tells how to rotate a font. This can be used for the global GDIInfo.
Though announcing a certain value not all fonts may be able to follow that flag.
There is another font specific flag in WindowsIfiInfo, which tells about the single device font.

The element provides the following attributes:
Angle (optional)
AntiClockwise (mandatory)
EscapementAffected (mandatory)

Angle

This attribute tells about the minimum angle for rotation.

AntiClockwise

It tells whether the PDL is rotating anti- or clockwise.

EscapementAffected

This is a Boolean attribute.
If the escapement is affected, it is not necessary to send a positioning command for the next character to print with the same angle again. See PCL 5e.

The element has the following sub-elements:
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.3.3.1.5.1. Parameter/param:ParameterX

See the Common Elements section for information about these elements.

3.3.3.3.1.6. DeviceColor

In this section some basic and global color parameters are defined.

This element tells about the colors a device is able to provide. Each color is specified by RGB values. Each device has at least one device color.

The element provides the following attributes:
Red (mandatory)
Green (mandatory)
Blue (mandatory)

Red

This field tells about the red part of the RGB value.

Green

This field tells about the green part of the RGB value.

Blue

This field tells about the blue part of the RGB value.

The element has the following sub-elements:
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.3.3.1.6.1. Parameter/param:ParameterX

See the Common Elements section for information about these elements.

3.3.3.3.1.7. GlobalTextCapabilities

This element defines some global font parameters.

The element provides the following attributes:
ShowBitmapFonts (mandatory)
ShowVectorFonts (mandatory)
CharacterOutputPrecision (mandatory)
CharacterClippingPrecision (mandatory)

ShowBitmapFonts

This is a Boolean attribute, which may be overwritten by the driver anyhow. True means to show bitmap system (!) fonts. Supposed to be to true in most cases.

ShowVectorFonts

This is a Boolean attribute, which may be overwritten by the driver anyhow. True means to show bitmap system (!) fonts. Supposed to be to true in most cases.

CharacterOutputPrecision

This is a Boolean attribute and tells about exact positioning of text in pixel units. Please check the Microsoft DDK for details.

CharacterClippingPrecision

This is a Boolean attribute. Please check the Microsoft DDK for details.

The element refers to the following sub-elements:
Scaling (mandatory, single)

3.3.3.3.1.7.1. Scaling

Global info about the scaling method of scalable fonts.
To be discussed that there cannot be two different scaling methods in one PDL.

The element provides the following attributes:
Anisotropic (mandatory)
Integer (mandatory)
Continuous (mandatory)
Doubling (mandatory)

Anisotropic

This is a Boolean attribute. True means Anisotropic, false means Isotropic.

Integer

This is a Boolean attribute. True means Integer, false means NonInteger. Check the Microsoft DDK, as this is some pretty old technology.

Continuous

This is a Boolean attribute. True means Continuous, false means NonContinuous.

Doubling

Predefined values: Doubling, NoDoubling
This is a Boolean attribute. True means Doubling, false means NoDoubling. Check the Microsoft DDK, as this is some pretty old technology.
This field tells whether the device can double a font in its width and height, but then has no other scaling method. A driver may always overwrite this value and never support doubling.

3.3.3.3.2. DeviceFont

This section contains the definition and description of single device fonts.

The element provides the following attributes:
ID (mandatory)
FontFamily (mandatory)
FontVendor (optional)

Encoding (mandatory)
IFIMETRICSSelection (optional)
PassiveFont (optional)

ID

It is recommended to assign a unique technical font id to each device font for future reference. A typical example would be to use the Font_ID for referring to a subfont.

FontFamily

This attribute specifies, which font family the font belongs to, following Microsoft's classification. You may learn about the constraints of the attribute in Microsoft's DDK or MSDN.

FontVendor

This attribute may hold information about the font vendor.

Encoding

A number of attributes require a Unicode number.
The only value supported so far is a two-byte Unicode. To be verified.

IFIMETRICSSelection

This attribute indicates certain font attributes with a simple bit combination.
Please refer to Microsoft's DDK for further details.

PassiveFont

If a font is declared passive it will not be announced actively to the operating system, but the driver will likely substitute any system font, which is available in the device.

The element refers to the following sub-elements:
FontNames (mandatory, single)
GlobalMetrics (mandatory, single)
IFIMETRICSInfo (mandatory, single)
Panose (optional, single)
FontTechnology (optional, single)
CharacterHandling (mandatory, single)
CharacterList (mandatory, single)
KerningTable (optional, single)
CharacterSubstitutionTable (mandatory, single)
PseudoDeviceFonts (mandatory, single)
FontReference (optional, multiple)
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.3.3.2.1. FontNames

This section tells about font strings used for the font itself as well as for some of its attributes.

As all elements are optional, you can make some strategic decisions how you want a certain font to be handled. This includes passive fonts as well as subfonts.
Passive fonts are not announced actively to the operating system. So they will not be listed as device fonts. But when a system font with the same name is used, it will be substituted with the correspondent device font by the driver.
Subfonts are extensions to other device fonts and are not a font by its own. They are referenced by another device font.

The element refers to the following sub-elements:
FaceName (optional, single)
FamilyName (optional, single)
UniqueName (optional, single)
AliasName (optional, multiple)
SlantName (optional, single)
WeightName (optional, single)

3.3.3.3.2.1.1. FaceName

The element provides the following attributes:
ID

ID

See the Common Attributes section for common information about this attribute.
This is considered the font name the user sees in all applications. In this case the developer intentionally may not resolve the reference in the locale, which would mean the reference string will be used.
If this field is omitted, the font is supposed to be a passive one. That means it will not be announced as a resident font to the operating system, but will replace the system font with the same name.
For some fonts this string is a combination of the family and the style. This is mostly the case, when there are more styles than the standard “Italic” and “Bold”.
Sample: Arial Narrow
Sample: Goudy Old Style Extra Bold

3.3.3.3.2.1.2. FamilyName

The element provides the following attributes:
ID

ID

See the Common Attributes section for common information about this attribute.
In this case the developer intentionally may not resolve the reference in the locale, which would mean the reference string will be used.
There is a field FamilyName in the IFIMETRICS structure, but they say it should be consistent with the facename in the logfont structure. See Microsoft DDK Windows NT.
Sample: Arial Narrow
Sample: Goudy Old Style

3.3.3.3.2.1.3. UniqueName

The element provides the following attributes:
ID

ID

See the Common Attributes section for common information about this attribute.
In this case the developer intentionally may not resolve the reference in the locale, which would mean the reference string will be used.
A more informative technical string. This parameter is a member of the IFIMETRICS structure. The authors of this documentation could not identify the use of this entry beyond doubt. It may be used for some vendor specific information in the font name.
Sample: Arial Narrow
Sample: Bitstream Swiss SWA 721

3.3.3.3.2.1.4. AliasName

Specify a list of font names (of type FaceName) other than the FaceName of this font, if you want those to be replaced by this font.
If this element together with the FaceName is omitted, the font is supposed to be a subfont referenced by another font and will not be announced as a resident font to the operating system.
The element provides the following attributes:
ID

ID

See the Common Attributes section for common information about this attribute.
In this case the developer intentionally may not resolve the reference in the locale, which would mean the reference string will be used.

3.3.3.3.2.1.5. SlantName

The element provides the following attributes:
ID

ID

See the Common Attributes section for common information about this attribute.
This is the string used for italic fonts and can be seen in the Font Style list box of applications.
Sample: Italic

3.3.3.3.2.1.6. WeightName

The element provides the following attributes:
ID

ID

See the Common Attributes section for common information about this attribute.
This is the string used for bold fonts and can be seen in the Font Style list box of applications.
Sample: Bold

3.3.3.3.2.2. GlobalMetrics

This section tells about global metrics parameters not caring about glyph specific values.

The element refers to the following sub-elements:
CharacterWidth (optional, single)

3.3.3.3.2.2.1. CharacterWidth

Please refer to the description of the element under element Character of element CharacterList.
At this location it specifies the character width of a fixed pitch font.

The element provides the following attributes:
UnitsPerEm (mandatory)
PointsPerInch (mandatory)
Spacing (mandatory)
MinimumHeight (mandatory)
MaximumHeight (mandatory)
HeightStep (mandatory)
Slant (mandatory)
WindowsWeight (mandatory)
WindowsAscender (mandatory)
WindowsDescender (mandatory)
MacAscender (mandatory)
MacDescender (mandatory)
MacLineGap (mandatory)
TypographicAscender (mandatory)
TypographicDescender (mandatory)
TypographicLineGap (mandatory)
AverageCharacterWidth (mandatory)
MaximumCharacterIncrement (mandatory)
CapHeight (mandatory)
xHeight (mandatory)
SubscriptXSize (mandatory)
SubscriptYSize (mandatory)
SubscriptXOffset (mandatory)
SubscriptYOffset (mandatory)
SuperscriptXSize (mandatory)
SuperscriptYSize (mandatory)
SuperscriptXOffset (mandatory)
SuperscriptYOffset (mandatory)
UnderscoreSize (mandatory)

UnderscoreOffset (mandatory)
UnderscoreAbsoluteValues (mandatory)
StrikeoutSize (mandatory)
StrikeoutOffset (mandatory)
StrikeoutAbsoluteValues (mandatory)
Baseline (mandatory)
Aspect (mandatory)
Caret (mandatory)
Orientation (mandatory)
Degree (mandatory)

UnitsPerEm

These are the design units for scalable fonts.
Generally for all fonts this is the cell height minus internal leading. This is the point size the user sees in applications.
The cell height is the sum of ascender plus descender.

PointsPerInch

Predefined values: 72, 72.2752
Intellifonts do not use 72 points per inch, but are following more traditional design rules.

Spacing

Predefined values: fixed, proportional
This attribute tells about the font spacing.

MinimumHeight, MaximumHeight, HeightStep

These three parameters when used would indicate that a device has limits with the font size and the steps it can increase in.
The expected unit is PointSize*100 (min height =8pt would result in a value of 800).
For bitmap fonts this value would be ignored.

Slant

If Null, the font is upright. The expected value is an angle in units of 3600 clockwise from the upright position.
This implementation only supports fixed values, not variable ones.

WindowsWeight

Numeric values in steps of 100.
See Microsoft DDK for reference.
This implementation only supports fixed values, not variable ones.

WindowsAscender

Supposed to be a positive value, calculated from the baseline to the upper edge of the cell.
Following a Microsoft specification this value is based on the maximum value of characters of a Western character set 1252.
For compatibility reasons this is often hardcoded.

WindowsDescender

Supposed to be a positive value, calculated from the baseline to the lower edge of the cell.
Following a Microsoft specification this value is based on the maximum value of characters of a Western character set 1252.
For compatibility reasons this is often hardcoded.

MacAscender

Supposed to be a positive value, calculated from the baseline to the upper edge of the cell.
This is based on Apple's specification.

MacDescender

Supposed to be a negative value, calculated from the baseline to the lower edge of the cell.
This is based on Apple's specification.

MacLineGap

This is the extra value added to the MacAscender and MacDescender to get a convenient line spacing on Apple systems.
As the external leading for Windows platforms is not explicitely listed here, it should be calculated by summing the three Mac values up and subtract the two Windows values.
In case the result is negative, convert to zero.

TypographicAscender

Supposed to be a positive value, calculated from the baseline to the upper edge of the cell.
This is supposed to be the typographic decision, where the baseline should be.

TypographicDescender

Supposed to be a positive value, calculated from the baseline to the lower edge of the cell.
This is supposed to be the typographic decision, where the baseline should be.

TypographicLineGap

For compatibility reasons over platforms the sum of the three Mac values should be the same as of the three typographic values.

AverageCharacterWidth

The average character width of a font.
It is recommended to calculate the average of all characters of a font, independent from any character set.
Values are in virtual units for bitmap fonts and in design units for scalable fonts.

MaximumCharacterIncrement

The maximum character width of a font.
It is recommended to announce the widest character of a font, independent from any character set.
Values are in virtual units for bitmap fonts and in design units for scalable fonts.

CapHeight

The character height of upper case characters of a font.

Values are in virtual units for bitmap fonts and in design units for scalable fonts.
For details see Microsoft DDK for reference.

xHeight

The character height of lower case characters of a font like an ‘x’.
Values are in virtual units for bitmap fonts and in design units for scalable fonts.
For details see Microsoft DDK for reference.

SubscriptXSize

Supposed to be a positive value in virtual units for bitmap fonts and in design units for scalable fonts calculated corresponding to the value in UnitsPerEm.

SubscriptYSize

Supposed to be a positive value in virtual units for bitmap fonts and in design units for scalable fonts calculated corresponding to the value in UnitsPerEm.

SubscriptXOffset

Supposed to be a positive value in virtual units for bitmap fonts and in design units for scalable fonts calculated corresponding to the value in UnitsPerEm.

SubscriptYOffset

Supposed to be a positive value in virtual units for bitmap fonts and in design units for scalable fonts calculated corresponding to the value in UnitsPerEm.

SuperscriptXSize

Supposed to be a positive value in virtual units for bitmap fonts and in design units for scalable fonts calculated corresponding to the value in UnitsPerEm.

SuperscriptYSize

Supposed to be a positive value in virtual units for bitmap fonts and in design units for scalable fonts calculated corresponding to the value in UnitsPerEm.

SuperscriptXOffset

Supposed to be a positive value in virtual units for bitmap fonts and in design units for scalable fonts calculated corresponding to the value in UnitsPerEm.

SuperscriptYOffset

Supposed to be a positive value in virtual units for bitmap fonts and in design units for scalable fonts calculated corresponding to the value in UnitsPerEm.

UnderscoreSize

Tells about the line thickness.
Supposed to be a value in virtual units for bitmap fonts and in design units for scalable fonts calculated corresponding to the value in UnitsPerEm.

UnderscoreOffset

Supposed to be a value in virtual units for bitmap fonts and in design units for scalable fonts calculated corresponding to the value in UnitsPerEm.
Value is positive for the expected offset in descender direction.

UnderscoreAbsoluteValues

This is a Boolean field.
Always TRUE for bitmap fonts. Tells whether the value is to be used as an absolute value (TRUE= point size independent) for scalable fonts.

StrikeoutSize

Tells about the line thickness.
Supposed to be a value in virtual units for bitmap fonts and in design units for scalable fonts calculated corresponding to the value in UnitsPerEm.

StrikeoutOffset

Supposed to be a value in virtual units for bitmap fonts and in design units for scalable fonts calculated corresponding to the value in UnitsPerEm.
Value is positive for the expected offset in ascender direction.

StrikeoutAbsoluteValues

This is a Boolean field.
Always TRUE for bitmap fonts. Tells whether the value is to be used as an absolute value (TRUE= point size independent) for scalable fonts.

Baseline

Specifies a vector (x,y) that points in the writing direction. (-1,0) would be right to left, (0,-1) would to top-down...
A standard Western font would show 1,0.

Aspect

Check the IFIMETRICS structure for use of this parameter.

Caret

Is a vector that should be used by an app to display the blinking cursor in a text editing mode so that it matches the slant of text. Is not very useful for printer fonts. If filled out should be identical to italic/slant definition.

Orientation

Tells, in which orientation this font is available.
Very much PDL specific.

Degree

Tells about the minimum rotation step anticlockwise of the device font in units of n/3600.
Very much PDL specific.
Although the driver has to announce a general info about rotation capabilities, it may be that not all of the fonts can follow that announcement. It is up to the driver to use fallback mechanisms.

If the driver can get some system font information about the missing rotations instead, fine.
If not, it might be possible to create a functionality to position the single characters according to a certain degree, while the character itself will not be rotated.
Every other solution would be even more dramatic.

3.3.3.3.2.3. Panose

It is recommended to specify the ten Panose font classification attributes for device fonts.
These attributes may be used for font substitution by the driver.
All constraints for the various attributes are taken from the Panose specification, which you can find at a number of places like the MSDN.

The element provides the following attributes:
PanoseFamilyStyle (mandatory)
PanoseSerifStyle (mandatory)
PanoseWeight (mandatory)
PanoseProportion (mandatory)
PanoseContrast (mandatory)
PanoseStrokeVariation (mandatory)
PanoseArmStyle (mandatory)
PanoseLetterform (mandatory)
PanoseMidline (mandatory)
PanoseXHeight (mandatory)

All Panose attributes refer to a predefined data type. As this section is supposed to be filled out by experts, we do not further elucidate those attributes.

3.3.3.3.2.4. FontTechnology

Only for experts!
This section tells how scalable fonts are rendered. It can be left out for bitmap fonts.

The element refers to the following sub-elements:
StandardRendering | ScalingFormula

3.3.3.3.2.4.1. StandardRendering

The driver should have functions to handle the most important standard rendering types. Together with the two other attributes the driver should be able to calculate the width of characters in pixel.

The element provides the following attributes:
RenderingType (optional)
FixedIntegerBits (required)
FixedFractionBits (required)

RenderingType

A reference to a small list of standard rendering types.

FixedIntegerBits

Fill in the proper value a function would need additionally to the rendering type.

FixedFractionBits

Fill in the proper value a function need additionally to the rendering type.

3.3.3.3.2.4.2. ScalingFormula

As this is one of the most complex fields, where a parameter converter is used, it is absolutely necessary to study the specification of the parameter converter first and to understand each and every detail. Next it is recommended to study some sample fonts already defined by other developers, if possible based on different rendering types. Additionally it is recommended to have a contact ready in case of an emergency.
Typical input parameters are the actual font height (DRV_FontHeight), the device resolution and the character width.
The expected result of the scaling formula is in positioning units (pixel).

3.3.3.3.2.5. CharacterHandling

The element provides the following attributes:
UserDefinedCharacterSet (optional)
NonBreakingSpace (mandatory)
DummyCharacter (mandatory)

UserDefinedCharacterSet

This attribute tells whether the font shall be realized with user defined character sets. Intended to be used especially for PostScript drivers. It defaults to FALSE.
The driver must know about the PDL specific identifiers for this implementation. It will find that information in the CharacterName attribute of element Character.
This spec is not specifying how a user defined character set is to be assembled. The driver has to know that by itself when detecting the PDL.

NonBreakingSpace

This attribute tells which character will be used for the non-breaking space.
The entry is encoding specific.

DummyCharacter

This attribute tells which character shall be used in case the required character cannot be realized.
The entry is encoding specific.

The element refers to the following sub-elements:
OSCharacterSet (mandatory, multiple)

3.3.3.3.2.5.1. OSCharacterSet

If the font could not support at least one operating system character set, it would be useless. So this element is mandatory. This section may grow over the time for several operating systems and subversions of it.

The element provides the following attributes:
ID

ID

This attribute is the operating system specific identifier of the host character set. This is not for telling about device character sets!
This spec is referring to the Unicode Consortium for the specification of these character sets. So a driver can learn which Unicode values each character set supports.
A driver is supposed to check, which locales are available on the host. These locales with the corresponding character sets are necessary to screen device fonts.
Depending on the host configuration certain device fonts may be activated or deactivated.

3.3.3.3.2.6. CharacterList

Each entry is a set of information for one character. All together it makes up the listing of all characters of a font sorted by the encoding.
You can find information about the width of every character as well as how each character can be printed.

The element refers to the following sub-elements:
Character (mandatory, multiple)

3.3.3.3.2.6.1. Character

This element describes how one character is to be handled.

The element provides the following attributes:
ID
CharacterPrintOutput (optional)
CharacterName (optional)

ID

This attribute identifies the character in the specified encoding (see DeviceFont attributes).

CharacterPrintOutput

This attribute tells what has to be sent to the printer additionally to the command sequences to get a certain character. In nowadays PDL this typically is one byte.
If left out, the ID of element Character is to be used for output as well. Eventually this is the preparation for a Unicode printer.

CharacterName

This attribute is prepared for PostScript support.
In case a PostScript driver wants to work with user defined character sets, it has to know about the PostScript names of the characters. These can be defined here.

The element refers to the following sub-elements:
CharacterWidth (optional, single)
SymbolSet (optional, single)
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.3.3.2.6.1.1. CharacterWidth

The width of a character, typical of a proportional font.
Supposed to be a value in virtual units for bitmap fonts (e.g. all three attributes sum up as 720 for a 10 cpi bitmap font with virtual units of 7200) and in design units for scalable fonts.
If left out, a fixed value is expected in the scaling formula of a scalable font. For bitmap fonts the fixed value can then be found in the description of the CharacterWidth element under GlobalMetrics.
The driver has to check values in subfonts. An Asian font, where all characters are fixed pitch, but some are half width, can be described as a fixed pitch with the half width characters described in a subfont. An Asian font with most of the characters as fixed pitch (some half width), but some are proportional, can be described as a proportional with the fixed pitch characters in the main font, the half width characters in a subfont and the proportional characters in another subfont.
This specification does not provide a mechanism for scaling in one direction. If there are several fonts with the same height, but different width, they must be specified as separate fonts.

The element provides the following attributes:
A_Value (mandatory)
B_Value (mandatory)
C_Value (mandatory)

A_Value, B_Value, C_Value

These three attributes determine the leading space, the actual character width and the trailing space of a character.

3.3.3.3.2.6.1.2. SymbolSet

The element has the following sub-elements:
Parameter (optional, single)
param:ParameterX (optional, single) or any other wildcard element

3.3.3.3.2.6.1.2.1. Parameter/param:ParameterX

See the Common Elements section for information about these elements.
This should refer to the parameter necessary to be sent to the printer to select a special symbol set. The driver should check the parameter for each character. If identical, don’t send again. The corresponding

command sequence is ruled by event DeviceFontCharacter. If left out for a character, don't send the command sequence.

3.3.3.3.2.6.1.3. Parameter/param:ParameterX

See the Common Elements section for information about these elements.
This should refer to the parameter necessary to be sent to the printer to define a special character ID as a printing character, while certain character codes may refer to escape codes by default. The driver should check the parameter for each character. The corresponding command sequence is ruled by event DeviceFontCharacter. If left out for a character, don't send the command sequence. While the parameters are not necessarily different (as a matter of fact, it may always be something like a '1'), this element works more like a boolean, whether the command sequence has to be sent or not. The command sequence is to be sent for each character, if the Parameter element is specified.

3.3.3.3.2.7. KerningTable

This element tells about kerning pairs as the only kerning method.

The element refers to the following sub-elements:
KerningPair (mandatory, multiple)

3.3.3.3.2.7.1. KerningPair

This element describes a kerning pair for one character.

The element provides the following attributes:
FirstCharacter (mandatory)
SecondCharacter (mandatory)
XOffset (mandatory)

FirstCharacter
This is the character ID in the specified encoding for the first character of the pair.

SecondCharacter
This is the character ID in the specified encoding for the second character of the pair.

XOffset
This is the horizontal offset (for Latin fonts) between the two characters.
This is supposed to be a negative value, the default in most cases, if the second character is overlapping the first one partly, and a positive to provide an extra space.
Supposed to be a value in virtual units for bitmap fonts and in design units for scalable fonts.

3.3.3.3.2.8. CharacterSubstitutionTable

This element contains the specification of character substitutions. It depends on the driver's font implementation, if it is using its own generic character substitution function or if it is looking for font specific information, which it can find here.
A character may be substituted by one other character or by a set of two character, which are printed more or less on top of each other.

The element refers to the following sub-elements:
CharacterSubstitution (mandatory, multiple)

3.3.3.3.2.8.1. CharacterSubstitution

This element describes a substitution for one character.

The element provides the following attributes:
ID
SubstitutionFirstCharacter
SubstitutionSecondCharacter (optional)
SubstitutionXOffset (optional)
SubstitutionYOffset (optional)

ID
This attribute identifies the character to be substituted, based on the specified encoding.

SubstitutionFirstCharacter
This attribute identifies the first substitution character, based on the specified encoding.

SubstitutionSecondCharacter
This attribute identifies the second substitution character, based on the specified encoding.
When a second character is used in the character substitution specification, it must not be wider or higher than the first substitution character.

SubstitutionXOffset
This attribute specifies the horizontal offset (for Latin fonts) for the second substitution character, calculated from the left edge of the first substitution character cell.

SubstitutionYOffset
This attribute specifies the vertical offset (for Latin fonts) for the second substitution character, calculated from the top edge of the first substitution character cell.

3.3.3.3.2.9. PseudoDeviceFonts

This section tells about certain attributes of a device font and refers to the modifying attributes.

When a certain style, weight, underline or strikeout mode is requested, the driver has to find out, if the device is implementing that function by transforming the original font, which would mean the font would refer to those modifying attributes, or if another font with those attributes has to be identified.
With printers typically attributes like underline and strikeout are realized by transformation. Attributes like style and weight vary a lot in different implementations.
When there is a reference to a modifying attribute the corresponding parameter in GlobalMetrics is redundant.
Modifying attributes are supposed to be an On/Off toggle. So you are not supposed to specify more than two records per attribute per font, although two different fonts theoretically can refer to different records (although this is considered very unusual).

The element refers to the following sub-elements:
PseudoStyle (0-2)
PseudoWeight (0-2)
PseudoUnderline (0-2)
PseudoStrikeout (0-2)

3.3.3.3.2.9.1. PseudoStyle

A reference to a slant of element Slant within the modifying attributes.
To provide a sample:
Imagine two modifying attributes for slant have been defined for upright and italic and one (or more) specific font can be transformed correspondingly. The description is supposed to show two elements of PseudoSlant with corresponding references. There have to exist two corresponding elements of element Slant within the modifying attributes to identify the command sequence to be sent to the device to realize the attribute properly.

3.3.3.3.2.9.2. PseudoWeight

A reference to a weight of element Weight within the modifying attributes.

3.3.3.3.2.9.3. PseudoUnderline

A reference to a record of element Underline within the modifying attributes.

3.3.3.3.2.9.4. PseudoStrikeout

A reference to a record of element Strikeout within the modifying attributes.

3.3.3.3.2.10. FontReference

Each reference points to another device font by listing its ID. These fonts will then be used as subfonts.

The element provides the following attributes:
ID

ID

This attribute is referring to the ID of another device font.

There is no problem with referring to a fixed pitch font under a proportional font. The ruling font determines the final parameter. But it may well save space and time to refer to a fixed pitch subfont for certain groups of characters. This is especially supposed to be used for Asian fonts.

3.3.3.3.2.11. Parameter/param:ParameterX

See the Common Elements section for information about these elements.

This is the parameter dealing with the command sequence to actually select the font.

This attribute refers to the command sequence file and specifies the command sequence to be sent to the device to get the required font.

For the work in the command sequence file please keep in mind:

Especially in this case it is recommended to work with the parameter attributes instead of putting all info into the command sequence attribute. This would allow to work with the same command sequence for many fonts.

Studying the parameter converter in detail is an absolute necessity. This may likely be one of most complex command sequences you ever enter.

The command sequence is considered redundant meaning that it is not necessary to repeat it every time, even when the font is selected already.

You may work with one parameter telling whether a font is proportional or not.

You may work with one parameter for the height of the font. For scalable fonts this would be a formula, which can either be handled in the parameter attribute (recommended) or directly in the command sequence attribute.

You may work with one parameter for the width of the font, typically used for fixed pitch fonts only. For scalable fonts this would be a formula, which can either be handled in the parameter attribute (recommended) or directly in the command sequence attribute.

You may work with one parameter telling whether the font is italicized.

You may work with one parameter telling whether the font is bold.

You may work with one parameter for the typeface of the font.

3.3.3.3.3. wildcards

See the Common Elements section for information about these elements.

This could connect to some kind of additional font related info, perhaps font download.

3.3.3.4. wildcards

See the Common Elements section for information about these elements.

May be this could be used to connect to some kind of vector object definition.

3.3.4. Dependencies

Historically dependencies are defined for user interface interactions.

They are checked when a dialog is initialized or when a feature in the driver's dialog is changed to check, if the change causes any problems for other features.
A driver's dialog is initialized either by its own settings of its last instance or - depending on the platform - the dialog may be opened within a standard application, which may overwrite certain parameters by application's settings.
Depending on the platform dependencies may as well be checked during printing, as applications may create interfering settings, which have not been selected in a driver's dialog, but in an application's dialog like Page Setup. So a driver cannot rely that it recieves clean settings from an application.

The specification allows the definition of simple dependencies as well as more complex ones.

The element refers to the following sub-elements:
Dependency (mandatory, multiple)

3.3.4.1. Dependency

Dependencies are supposed to be checked in the order they are defined.
A user policy is to be checked after a unit description and therefore has higher priority.

If no action is defined, the driver will ignore the dependency.
The three possible actions are considered mutually exclusive.

The element refers to the following sub-elements:
OperatingSystem (optional, multiple) | FeatureCondition (mandatory, single) | Action (mandatory, single)

3.3.4.1.1. OperatingSystem

UPDF is operating system independent, but platforms may behave differently.
So we provide a chance to specify OS dependencies.
If not used the user dependency is considered global, which should be the general case.

This entry offers a list of platforms.

3.3.4.1.2. FeatureCondition

As many features as required can be listed, which represent an 'AND'.
Eventually it works like 'if this is true and this is true and this is true ...'.
There is no 'OR'. This was specified in an earlier version, which made the dependencies more compact. We have learnt during the implmentation of a standard in drivers, that parsing would get really complicated. It would make development very complex and likely slow down the performance. So we give up on that. An 'OR' has to be represented by a new dependency.

The element provides the following attributes:
FeatureID
ConditionOperator

FeatureSetting

FeatureID

This is an attribute of type string, as it may be referred to from other instances than the device description itself (e.g. the user policy instance).
It refers to a feature.

ConditionOperator

The description developer can choose between settings 'equal' or 'not-equal'. We think that all definitions can be covered by this choice.

FeatureSetting

3.3.4.1.3. Action

The element provides the following attributes:
DominantFeature (optional)

DominantFeature

While the FeatureCondition lists all features involved in the condition, an action can be specified depending on which of the involved feature is currently selected.
This architecture is chosen to keep the dependencies more compact. An alternative could have been to design separate dependencies, if different actions are required depending on the feature selected.
If no dominant feature is specified, the Action is to be performed unconditional.

It is recommended to study the use of a dominant feature under a composite feature, too.

At least one action has to be specified.

The element refers to the following sub-elements:
ActionFilter | ActionMessage | ActionSelection (mandatory, multiple)

3.3.4.1.3.1. ActionFilter

If the element is selected, its setting is ‘true’.
That means the driver/client is supposed to filter the specified record of the specified DominantFeature, if element ActionFilter is set.

3.3.4.1.3.2. ActionMessage

This element is used, when a message should pop up under a certain condition.
This may (not necessarily) be used, if the driver offers all entries of the feature for selection and the user will be warned about dependencies once a 'bad' selection is made.

The element provides the following attributes:

ID

ID

See the Common Attributes section for common information about this attribute.
This attribute refers to the message text, which should pop up in the message dialog.

The element refers to the following sub-elements:
ActionMessageOK | ActionMessageCancel (1 - 2)

3.3.4.1.3.2.1. ActionMessageOK

If this element is available, the driver is supposed to support an 'OK' button within the message.
The SetFeature element defines how the driver should react when the user pushes the OK button.

The element provides the following attributes:
ID

ID

See the Common Attributes section for common information about this attribute.
The attribute is the identifier for the text for the button.

The element refers to the following sub-elements:
SetFeature (mandatory, multiple)

3.3.4.1.3.2.1.1. SetFeature

The element is explained under the element ActionSelection.

3.3.4.1.3.2.2. ActionMessageCancel

If this element is available, the driver is supposed to support a 'Cancel' button within the message.
When the Cancel button is pushed, the driver should reject the currently selected setting and re-select the original setting.

The element provides the following attributes:
ID

ID

See the Common Attributes section for common information about this attribute.
The attribute is the identifier for the text for the button.

3.3.4.1.3.3. ActionSelection

The third type of action, which can be specified for dependencies are selections.
Selections, which are known from element ActionMessageOK already, can be used as the only action.

The element refers to the following sub-elements:
SetFeature (mandatory, multiple)

3.3.4.1.3.3.1. SetFeature

The element provides the following attributes:
Appearance (default: standard)
FeatureID
FeatureSetting

Appearance

See the Common Attributes section for common information about this attribute.
Once a feature is set to a specific record, it may be grayed.

FeatureID

This is an attribute of type string, as it may be referred to from other instances than the device description itself (e.g. the user policy instance).
It refers to a feature.

FeatureSetting

The element has the following sub-elements:
InfoButton (optional, single)

3.3.4.1.3.3.1.1. InfoButton

Some drivers are capable of warning the end user that the setting has been changed and offer information to find out why. This is typically something like an Info button.
This specification does not reference a picture to be used to indicate the availability of additional information. It is left to the implementation of the driver to use the text referred to by the ID or replace it with a picture.

The element provides the following attributes:
ID

ID

See the Common Attributes section for common information about this attribute.
The attribute is the identifier for the text for the button.

The element refers to the following sub-elements:
InfoButtonOK (mandatory, single)

3.3.4.1.3.3.1.1.1. InfoButtonOK

If this element is available, the driver is supposed to support an ‘OK’ button within the message.

The element provides the following attributes:
ID

ID

See the Common Attributes section for common information about this attribute.
The attribute is the identifier for the text for the button.

Some idea how to use SetFeature:
You can realize something like 'if record1 of feature1 is selected, record 4 of feature11 shall be selected and feature11 should be grayed out.

3.3.5. Events

The element refers to the following sub-elements:
Event (mandatory, multiple)

3.3.5.1. Event

Each event element refers to one command sequence and tells when and how it is to be handled.

The element provides the following attributes:
ID
EventCommandPosition
EventID
Redundant (optional)
CommandSequenceID

ID

See the Common Attributes section for common information about this attribute.
This is the unique identifier of the special event.

EventCommandPosition

This attribute tells, whether a command sequence is to be sent at the start of the end of a special event. Cuts down the list in half.

EventID

This attribute is the selected event from a predefined list like ‘Job’, ‘Document’, ‘Media’, etc.

Redundant

If a command sequence is declared redundant, it does not have to be sent as long as the setting doesn’t change.

CommandSequenceID

This is the reference into the command sequence instance.

The element refers to the following sub-elements:
PreCondition (optional, multiple)
PostCondition (optional, multiple)
Reset (optional, multiple)
OptionalUnitEventLocation (optional, single)

3.3.5.1.1. PreCondition

The element provides the following attributes:
FeatureID
FeatureSetting

FeatureID

This is an attribute of type string, as it may be referred to from other instances than the device description itself (e.g. the user policy instance).
It refers to the ID of a feature.

FeatureSetting

This attribute refers to the ID of a feature record.

3.3.5.1.2. PostCondition

The element provides the following attributes:
FeatureID
FeatureSetting

FeatureID

This is an attribute of type string, as it may be referred to from other instances than the device description itself (e.g. the user policy instance).
It refers to the ID of a feature.

FeatureSetting

This attribute refers to the ID of a feature record.

3.3.5.1.3. Reset

This element refers to a feature.
It specifies, which feature is reset by this command sequence (and therefore may have to be set again).

3.3.5.1.4. OptionalUnitEventLocation

This element will not appear in the device description of the base unit, but likely in the one for optional units.

The element provides the following attributes:
Location
EventIDRef

Location

This attribute tells whether the event of the optional unit shall be inserted before or after the referred event.

EventIDRef

This is the identifier of the event reference, typically listed in the device description of the base unit (theoretically it could be listed in the unit description of an optional unit, too).
In case several events in different optional units refer to the same event (likely in the base unit description), the order is determined by the order the optional units are assigned to the device configuration.

☛ This could be a sample Job:
Imagine a 14 page job, with duplex and 6-Up settings. Pages 1 and 13 should be checked at start of media by the driver. Pages 1, 7 and 13 should be checked at start of physical pages. All pages should be checked at start of logical pages.

Most events show the same structure.
An event always has has an ID and an EventID with an EventCommandPosition attribute. The attribute Redundant and the three elements PreCondition, PostCondition and Reset are optional.
The Redundant attribute tells whether a command sequence needs to be sent in every occurence of an event even when the setting is unchanged.
We want to be able to set preconditions and postconditions per Event element. Here you can force feature to a specific setting for a certain event.
You can specify that certain Event IDs are to be reset after a special Event ID. In that case add at least one Reset element and set the ID of the required feature.

3.3.6. UserInterfaces

UserInterfaces is an optional element, but should prove to be a useful extension of a device description. It can be ignored by a driver in case it has its own idea how to arrange controls. The fallback for a driver expecting but not finding a user interface description is supposed to be something like a TreeView.
Any user interface specification could be realized as a TreeView, but there is reasonable information for more elaborate and fancier layouts.
Any information about screen resolution, pixels, size, justification and detailed appearance of a user interface control is considered out of range of this specification, as this may depend on platform specific implementations. However there should be sufficient information for various implementations.

The element refers to the following sub-elements:
UserInterface (mandatory, multiple)

3.3.6.1. UserInterface

The element provides the following attributes:
ID
HelpID (optional)
UIComposition
UIPosition (optional)

ID

See the Common Attributes section for common information about this attribute.

HelpID

The HelpID is an identifier as well as the ID and therefore a candidate to be localized.
This attribute holds the text string for the context sensitive help.

UIComposition

This attribute is explained in element UIGroup.

UIPosition

This attribute is explained in element UIGroup.

The element refers to the following sub-elements:
OperatingSystem (optional, multiple)
LocaleRef (optional, multiple)
UIFeature (optional, multiple)
UIGroup (optional, multiple)

3.3.6.1.1. OperatingSystem

If user interfaces are specified in a device description it is expected that there is one user interface available without an OperatingSystem element. This is the candidate for the fallback, if a specific platform is not supported.
Additional user interfaces each with a single OperatingSystem element set to a specific platform can be specified to support a different set of controls and/or to arrange them in a different way.

3.3.6.1.2. LocaleRef

By default you may design on user interface for all locales.
Later it may come in handy to have the chance to assign special locale references to some user interfaces. It is recommended to be careful with using that capability not to create too many combinations of operating systems and locales.

3.3.6.1.3. UIFeature

This element is explained in the UIGroup element.

3.3.6.1.4. UIGroup

The element provides the following attributes:
ID
HelpID (optional)
UIComposition
UIPosition (optional)

ID

See the Common Attributes section for common information about this attribute.

UIComposition

This attribute indicates, if the user interface should open another level of depth. If set to juxtaposed, all controls are supposed to be arranged on the same level. If set to superimposed, the driver is supposed to

hide the underlying controls in a deeper UI level (like a subdialog). If set to page-based, the driver is supposed to arrange the underlying controls in pages, e.g. on tabs.
If the ID of a group is resolved in the locale and the Composition is set to juxtaposed, the corresponding string is supposed to be used as the group's label. If the ID of a group is resolved in the locale and the Composition is set to superimposed, the corresponding string may be realized in a button control. The text string of the resolved ID will likely be ignored in a page-based composition.

UIPosition

The Position attribute looks like a cell position in a spreadsheet with a combination of a row and a column parameter (row,column).
This allows for positioning of a control within a higher level group of controls, e.g. to position a feature in the third row and the second column of a group of features.

The element refers to the following sub-elements:
UIFeature (optional, multiple)
UIGroup (optional, multiple)
wildcards (optional, single)

3.3.6.1.4.1. UIFeature

The element provides the following attributes:
FeatureID
UIPosition (optional)

FeatureID

This attribute refers to one of the device description features. To be specific: it refers to the ID attribute of an element under Features.

UIPosition

This attribute is explained in element UIGroup.

The element refers to the following sub-elements:
wildcards (optional, single)

3.3.6.1.4.1.1. wildcards

This is a chance to refer to another schema, which could further specify the user interface.

3.3.6.1.4.2. UIGroup

You see that this element can be used recursively.

3.3.6.1.4.3. wildcards

This is a chance to refer to another schema, which could further specify the user interface.

3.3.7. wildcards

See the Common Elements section for information about these elements.
This wildcard may be used, if somebody wants to add other device capabilities than PrintCapabilities, e.g. ScanCapabilities.

Command Sequences Schema

The corresponding schema is named UPDFCommandSequences and can found at http://www.pwg.org/schemas/upd/1.0/UPDFCommandSequences.xsd.

It's best to think of command sequences in four steps.
The first step is to specify when which command sequence should be sent. This is done in the events section of the unit description and only involves a reference to the real sequence.
As the second step this reference gets resolved in the command sequences schema, which shows the real command sequence to be sent to the device in the attribute CommandSequenceID. But in many cases this term still shows a parameter, which has to be replaced depending on the actual setting of that feature.
If the command sequence involves a parameter, the driver has to check the actual setting and identify the proper parameter in the according feature record. This is step 3.
As step 4 that parameter gets resolved in the second section ResolvedParameters of the command sequence instance.
This architecture keeps the unit description free of any job and page description language commands.

4. Commands

This is the top element of the Command Sequences schema.

The element provides the following attributes:
FileTypeIdentification

FileTypeIdentification

The FileTypeIdentification attribute shows "Command sequences and parameters".

The element refers to the following sub-elements:
FileInfo (optional, single)
CommandSequences (optional, single)
ResolvedParameters (optional, single)

4.1. FileInfo

A description developer may save some information about the actual instance.

The element provides the following attributes:
FileVersion

FileVersion

This is the only mandatory attribute.
Follow the typical pattern for a version (like UPDFVersion).

The element has the following sub-elements:
Comment (optional, multiple)

4.1.1. Comment

You can set as many other version parameters as you want.

4.2. CommandSequences

The element refers to the following sub-elements:
CommandSequence (mandatory, multiple)

4.2.1. CommandSequence

This element resolves a reference from an event in the unit description.

Explain variable values.

The element provides the following attributes:
CommandSequenceID
CommandSequence

CommandSequenceID

This attribute refers to a CommandSequenceID in the Event element of a unit description.

CommandSequence

The command sequence gets resolved here.
This attribute really describes what is to be sent to the device. It is assumed that the Parameter Converter will be heavily used at this place. So it is seriously recommended to study that section towards the end of this specification document.

4.3. ResolvedParameters

The element refers to the following sub-elements:
ResolvedParameter (mandatory, multiple)

4.3.1. ResolvedParameter

The element provides the following attributes:
Parameter
Value

Parameter

This attribute refers to the parameter ID, which was found in the setting of the actual feature record.

Value

This attribute shows the real parameter to be inserted into the command sequence.

Locale Schema

The corresponding schema is named UPDFLocale and can found at http://www.pwg.org/schemas/upd/1.0/UPDFLocale.xsd.

5. Locale

This is the top element of the Locale schema.

The element provides the following attributes:
FileTypeIdentification

FileTypeIdentification
The FileTypeIdentification attribute shows "Locale".

The element refers to the following sub-elements:
FileInfo (optional, single)
LocaleIdentifier (mandatory, single)
LocaleElements (optional, single)
LocaleDefaults (optional, single)

5.1. FileInfo

A description developer may save some information about the actual instance.

The element provides the following attributes:
FileVersion

FileVersion
This is the only mandatory attribute.
Follow the typical pattern for a version (like UPDFVersion).

The element has the following sub-elements:
Comment (optional, multiple)

5.1.1. Comment

You can set as many other version parameters as you want.

5.2. LocaleIdentifier

The element refers to the following sub-elements:

Language (mandatory, single)
Country (mandatory, single)

5.2.1. Language

The element provides the following attributes:
ClassifyingID

ClassifyingID

See the Common Attributes section for common information about this attribute.

5.2.2. Country

The element provides the following attributes:
ClassifyingID

ClassifyingID

See the Common Attributes section for common information about this attribute.

5.3. LocaleElements

The element refers to the following sub-elements:
LocaleElement (mandatory, multiple)

5.3.1. LocaleElement

The element provides the following attributes:
StringIDRef
LocalizedString

StringIDRef

This attribute refers to an ID attribute in the related unit description.

LocalizedString

This attribute resolves the ID and provides the localized text string.

5.4. LocaleDefaults

The element refers to the following sub-elements:
LocaleDefault (mandatory, multiple)

5.4.1. LocaleDefault

The element provides the following attributes:

DefaultFeature
DefaultSetting

DefaultFeature

This attribute refers to an ID attribute of a feature in the related unit description.

DefaultSetting

This attribute refers to an ID attribute of a feature record in the related unit description. For feature with variable values this will be an integer instead of an ID.

User Policy Schema

The corresponding schema is named UPDFUserPolicy and can found at
http://www.pwg.org/schemas/upd/1.0/UPDFUserPolicy.xsd.

The User Policy schema is an exact, but stripped down copy of the master schema.
Please study the elements and attributes used in the user policy schema in the master schema, as they are used identically.

6. DeviceCapabilities

This is the top element of the User Policy schema.
Notice that this is similar to the master UPDF schema.

The element provides the following attributes:
FileTypeIdentification

FileTypeIdentification
The FileTypeIdentification attribute shows "User Policy".

The element refers to the following sub-elements:
FileInfo (optional, single)
PrintCapabilities (mandatory, single)

6.1. FileInfo

A description developer may save some information about the actual instance.

The element provides the following attributes:
FileVersion

FileVersion
This is the only mandatory attribute.
Follow the typical pattern for a version (like UPDFVersion).

The element has the following sub-elements:
Comment (optional, multiple)

6.1.1. Comment

You can set as many other version parameters as you want.

6.2. PrintCapabilities

The element refers to the following sub-elements:
Features (optional, single)
Dependencies (optional, single)
UserInterfaces (optional, single)

6.2.1. Features

The element refers to the following sub-elements:
CompositeFeature (optional, multiple)

6.2.1.1. CompositeFeature

This is a straight reference to the CompositeFeature element in the master UPDF schema.

6.2.2. Dependencies

This is a straight reference to the Dependencies element in the master UPDF schema.

6.2.3. UserInterfaces

This is a straight reference to the UserInterfaces element in the master UPDF schema.

Some Q&A to elucidate the design.
Question: What exactly is the reason for user policies to exist?
Answer: While an IHV is supposed to provide the main device description, the system administrator or operator has the chance to add one user policy.
One of the main reasons for this architecture is the requirement that the main device description will never be changed again once created by the IHV to prevent any operators from damaging the main UPDF component. Therefore UPDF offers a method to add information for the driver, which is determined by customer needs. A system administrator can create exactly one user policy per UPDF device description instance.
The model in vision is that he/she will create as many directories as there are users or user groups, who will use the same device description, but with different pre-settings. One user policy per directory would be created and added to the corresponding device configuration file.
☛ Don't forget to add the user policy reference to the device configuration.

As you can see in the device configuration schema you can add new locales to the UserPolicy element as well.
You could even add additional locales without assigning a user policy description.
You would extend an original device configuration locale, if you want to provide different entries, or add new locales for new language/country sets.

A driver/client is expected to understand a user policy as an extension of the master device description and carefully merge the additional information to it.
Let’s imagine the Sales department as a user group, where a system administrator wants to provide special driver settings.

Different tasks one may want to accomplish with user policies:

New defaults
Defaults are managed in the Locale schema. Feel free to leave out LocaleElements. Define as many or as few new defaults as required per locale and add the new locales to the device configuration. The driver is supposed to check user policy defaults, as they can override the IHV defaults.
Sample: Change the media type default from ‘Standard Paper’ to ‘High Quality Paper’.

New user interface strings
UI strings are managed in the Locale schema as well. This is the same process as with defaults. Just use LocaleElements.
Be conservative with the length of strings.
Sample: Change the media type label from ‘Media Type’ to ‘Media for Sales’ and the entry ‘High Quality Paper’ to ‘HQ Paper 100g/m2’.

Hide entries in a UI control
You cannot create new entries in a UI control, but are able to hide existing ones.
Create a dependency to do that.
Sample: You may want to hide the media type ‘Copy Paper’ to prevent anybody in Sales from printing on it.

Create new UI controls
This is the top level of user policies and shows how much power a system administrator really has by deploying all user policy capabilities.
You cannot create standard features the IHV has not established previously, but you can build composite features on top of them.
Sample: The IHV device description may show single controls for media size, source and type. A system administrator may want to build one common Media control to handle all those attributes. This certainly needs some familiarity with the UPDF concept.
Some recommendations: You will make media size the dominant feature. In the sample used above you could also hide the complete media source control to only allow selection by media type. Be sure you select ‘AutoSelect’ as media source in all CompositeFeatureRecord elements.

For the expert: Let’s change the sample a bit. Now media size is not part of the composite feature, only source and type are. You want to keep all sources and all types as entries in a merged list maintained as the new media source list. That means you have to make media source the dominant feature and to override the old media source levels. You want to create as many composite feature records as there are source and type records (5 sources + 8 types would make up 13 c.f. records). One of the source records and one of the type records should be something like ‘AutoSelect’ or you could create dummies for the same purpose (these records are not counted in the 5 + 8). The new composite feature be reported to the operating system as the new media source (as that is the dominant feature!).

☛ The driver should be prepared to find records with the same ClassifyingID in the source records and react flexibly. You may create new classifying IDs on-the-fly where necessary.

One could drive composite features to the ultimate extreme by defining a c.f., which manages all other UI controls in the background, and hide all others. This control might be labelled ‘User’ and entries may look like ‘Marketing’, ‘Sales’ or ‘HR’. There is nothing wrong with that. Should work. You will very likely not specify a dominant feature in this sample, as there are normally a number of components in a driver, which have to reported to the operating system and its applications via APIs. You cannot substitute all those components with the one and only composite feature. So this sample is more working similar to the definition of a preset.

Laying out a new user interface

User policies allow for new user interfaces. This is an issue when working with composite features, as you create new UI controls and may want to hide others.

Some recommendations:

A user policy user interface replaces the corresponding user interface in the master device description.

☛ Don’t forget to check the Operating System entry.

The chance of ruining the description is pretty high. So this should be done by an absolute expert. The easiest way is to copy the whole user interface from the master description and then carefully modify it.

If a driver finds new UI controls in a user policy, but no new user interface, it should react flexibly. Old controls, which are supposed to disappear as a non-dominant feature of a c.f. with a dominant one assigned, should be expunged, but the location of others should not be affected by the action. Be careful how you add a new control. It might well result in a new page (e.g. a tab under MS Windows) of your user interface.

Prepare some mechanism for UserExtensible c.f. Something like an edit button next to it or a completely separate area for c.f. are just two possibilities. The user needs a chance to make the changes, but the layout and guise of the standard user interface should not be changed too much by that.

A composite feature without a dominant feature is a new UI control. A careful developer (system administrator) of a user policy would provide a new user interface. Prepare some fallback for the worst case like adding a new page to the user interface.

Things are pretty straightforward, if a new UI is provided and the design is done carefully.

Try to have a driver capable to read UPDF descriptions next to you when you develop user policies. So you can have it help you test your design.

The Parameter Converter

What is it?
When entering command sequences, some PDLs require binary values. Sometimes the output has to be formatted a certain way.
The Parameter Converter provides a special syntax to define terms with extraordinary requirements.
The basic idea of it is to be able to specify any command sequence while perpetuating human readability.
A driver/client reading a UPDF device description and especially a command sequence instance has to be able to interpret these terms and convert them properly.

History of a Parameter Converter
This specification of the Parameter Converter is based on the one used by Oak Technology, developed the first time at GCA mbH, Freiberg/N. Germany, in the early nineties. This is an extended version incorporating the experiences and increasing requirements over the years. The specification of the Parameter Converter as currently used by Oak Technology is declared public and is a contribution of Oak Technology to the UPDF group.

The Parameter Converter is case sensitive.
Do not use the following characters in ASCII, if you want to use them as a character and not with its special meaning in the Parameter Converter's syntax:
"%", "+", "-", "*", "/", "!", "<", ">", "(", ")", space
Sample: We recommend using a "%H" expression instead like "%H(37)", while "37" is decimal, instead of "%".
The percent character is used as the magic character.
"+", "-", "*", "/", "!", "<", ">" are used as operators.
The parentheses are used to show boundaries of an expression. It improves readability of a term.
The space character is used as a separator and will be ignored. It is recommended to use it regularly to improve readability.
"%A<bytes>" directly followed by an expression wrapped by left and right parenthesis identifies an ASCII format string. The parentheses don't count as bytes.
<bytes> identifies the maximum number of bytes expected. <bytes> is mandatory. A "0" indicates to use as many bytes as necessary. Imagine a variable keyword (like DRV_HorPos) and a "0" does not look that unusual any more.
"%F"<digits> directly followed by an expression wrapped by left and right parenthesis identifies an ASCII format string with <digits> number of digits after the decimal point. The part of the value in front of the decimal point is treated as if specified by "%A0".
"F" shall be the reminder to a kind of a floating point calculation.
"%H<bytes>" directly followed by an expression wrapped by left and right parenthesis identifies a binary format string with highest byte first.
<bytes> identifies the number of bytes expected. <bytes> is mandatory. A "0" indicates to use as many bytes as necessary.
"%L<bytes>" directly followed by an expression wrapped by left and right parenthesis identifies a binary format string with lowest byte first.

<bytes> identifies the number of bytes expected. <bytes> is mandatory. A “0” indicates to use as many bytes as necessary.
“%K” directly followed by an expression wrapped by left and right parenthesis identifies a predefined key.

There are two kinds of keys:
If the complete key starts with ‘DRV_’, it identifies a public key fed into the command sequence by the driver. See list of public driver keys in the table below. As these public keys will be used as parameters, the driver will pass in the proper value for a key.

Public key	**Meaning**
DRV_HorPos	dpi in PrinterResolution
DRV_VertPos	dpi in PrinterResolution
DRV_PageWidth	mm/1000
DRV_PageLength	mm/1000
DRV_RasterWidth	Actual raster resolution
DRV_RasterHeight	Actual raster resolution
DRV_ImageBytesActualScanline	Compressed, depending on plane implementation, count parameter not included
DRV_ImageBytesActualBand	Compressed, depending on plane implementation, count parameter not included
DRV_FontHeight	hundredth of pt, where 72pt = 1 inch

Table 1: Public driver keys

With the prefix ‘DRV_’ missing the key identifies an attribute or element in the schema. The attribute/element is to be replaced by the content of that attribute/element under the actual settings. The attribute/element does not have to be described with the full path beginning with DeviceCapabilities, but it must be possible to uniquely identify it. A dot ‘.’ works as a separator between levels of elements.
A setting of a variable value is represented by the element VariableValue together with the higher level elements to make it unique, e.g. ‘Copies.VariableValue’.

“%I” is directly followed by an expression wrapped by left and right parenthesis holding the condition of an IF statement and is followed by another expression wrapped by left and right parenthesis providing the term for the ‘true’ case.
Allowed operators for the first expression are “=”, “!=”, “<”, “>”, “<=”, “>=”.
“%E” is directly followed by an expression wrapped by left and right parenthesis is providing the term for the false case (%E is treated as a reminder to ‘else’). The %E statement is optional.
Nesting is possible, but be very careful. You have to work with parentheses smartly!
Two samples with different results:

```
    %I(a=1)(%I(b<5)(DoSomething) %E(DoSomethingElse))
    %I(a=1)(%I(b<5)( DoSomething)) %E(DoSomethingElse)
```

Several samples to practice:

The value before the colon represents the input of the driver.
The term in front of the arrow represents the snippet of the command sequence, the term after the arrow the result (some hexadecimal values in this sample) of the interpretation.

	%H1(27)	→	1Bh
DRV_HorPos = 258:	%H2(%K(DRV_HorPos))	→	01h02h
DRV_HorPos = 258:	%L2(%K(DRV_HorPos))	→	02h01h
DRV_HorPos = 48:	%L2(%K(DRV_HorPos))	→	00h30h
DRV_HorPos = 258:	%L1(%K(DRV_HorPos))	→	02h
DRV_HorPos = 58:	%A0(%K(DRV_HorPos))	→	58
DRV_HorPos = 58:	%A1(%K(DRV_HorPos))	→	5
DRV_HorPos = 58:	%A3(%K(DRV_HorPos))	→	058
DRV_HorPos = 5:	%F2(%K(DRV_HorPos)/2)	→	2.50

Imagine there is generic feature with ID GraphicMode in the instance to switch between Raster and GL/2, a sample for an IF statement to select some identifying font attributes for Courier New in PCL 5e could look like:

```
%I (%K(GenericFeatureRecord.ClassifyingID)=VectorMode)
(SD2,0,3,%F2(%K(DRV_FontWidth)),4,%F2(%K(DRV_FontHeight)),5,0,6,0,7,4099SS)
 %E (%H1(27) %H1(40) s %F2(%K(DRV_FontHeight)) hpsb4099T)
```

Author

Norbert Schade
norbertschade@comcast.net