Anvil: VDP Segmented Workflow Toolset

Fabio Giannetti, Royston Sellman
Digital Media Systems Laboratory
HP Laboratories Bristol

HPL-2007-18
February 7, 2007*

digital press,
indigo, VDP,
workflow, PPML,
PPMLT

* Internal Accession Date Only   Approved for External Publication

HP White Paper

Anvil: VDP Segmented Workflow Toolset

Release Version 1.01

Authors:
Fabio Giannetti
(fabio.giannetti@hp.com)
Royston Sellman
(royston.sellman@hp.com)

Table of Contents

List of figures

Overview

In recent years the publishing industry has witnessed a revolution. The introduction of Digital Presses has sent shock waves through the publishing industry, introducing new concepts and dilemmas. This revolution has similarities with the shift from analog to digital photography. The flexibility introduced by digital cameras has quickly won the hearts and minds of millions of consumers, even though the cost of printing digital pictures is higher. It has also opened up new applications, like photo sharing, photo kiosks and so on. In the digital publishing industry, we believe the same thing is happening, albeit at a slower pace.

Printing with digital presses increases flexibility. In fact, it is possible to generate very short runs and "every page is different" publication types. Moreover, several economic advantages are introduced by the digital press. For short run jobs, cheaper, faster plate imaging and less paper waste more than compensate for the higher cost per page compared to offset.

The industry is reacting to this revolution but far more slowly than consumers. A few changes are obvious:

- conventional Print Service Providers (PSP) are adopting digital to ensure they have both platforms
- new fully digital PSPs are entering the market to address the new opportunities
- consolidated digital PSPs are adding offset to increase their market share

In this complex and ever-changing scenario we believe that a profound change in the overall view of the publication industry is required. Personalization, customization, tailoring: these are the new features demanded by customers. However, printing 1,000 personalized pages with the same costs and quality as 1,000 identical pages can only be achieved if the personalization is generated semi-automatically and yet preserves the intent of the Graphic Artist (GA). We will defeat the object if we develop personalization technologies which are efficient but which lower the quality of the product.

This digital print revolution began with printing equipment but is quickly affecting the entire printing workflow and with it the respective actors. Again, the industry has responded in different ways. There are GAs who are slowly familiarizing themselves with Variable Data and Content Driven concepts. Some enterprises are reluctant to provide sensitive personal information to either GAs or untrusted PSPs and would like to control the data merging stage. Many PSPs have added new data processing capabilities to turn static documents into Variable Data Print (VDP) templates.

Nowadays, these interactions between the different actors are the real grey area of the publishing workflow. A proliferation of tools and formats has made this interaction even more difficult, and almost every workflow is a point solution. In recent years, new open formats have been introduced to tackle this issue. The most significant has been created by the Print On Demand initiative (PODi) and is called Personalized Print Markup Language (PPML). Several market leading companies gathered together under PODi to develop workflow interoperability solutions. Unfortunately their efforts have been undermined by a slow move to compliance and the proliferation of market differentiators, which usually translates into proprietary sub-formats.

Anvil addresses these issues with a set of software tools that make up a flexible, modular platform for Variable Data Print (VDP) workflows. Anvil workflows originate in creative tools familiar to graphic artists and designers, then pass through data-manipulation and merging tools targeted at business professionals, and then move on to a flexible and efficient composition/rendering engine capable of targeting the Indigo digital press particularly effectively. The Anvil project has an approach characterized by a few key principles:

- The platform has to be usable by professionals working in the field and must leverage their existing skill-set wherever possible. Thus the creative tools run as extensions to the popular layout design applications Quark XPress and Adobe InDesign; the data-manipulation stages run on WYSIWYG-style applications that offer marketing and business-modeling specialists a very high level of usability; the composition and rendering tools produce standards-based PPML output that is familiar to digital press operators
- Anvil is an end-to-end solution. From creation to the press, we have ensured that innovation in any single tool is supported throughout the workflow
- The VDP Template (the layout document that has some of its image and/or text elements made variable) is a key place to innovate. We have found that Anvil's relatively sophisticated templates greatly assist application to real VDP problems, indeed they simplify many of these problems
- VDP documents should be based on open standards. In Anvil the VDP template, and the template augmented with data rules and the data itself are represented in PPML-T, with actual variable text content described in XSL-FO. The final job for the press is written in PPML. In the case of jobs destined for the Indigo press the XSL-FO is composed into SNAP compatible SVG[1]. Open standards allow the use of a wide range of open software tools, reducing development costs and ensuring continued links to the mass testing and debugging characteristic of open software
- Composition of VDP documents should be left to the last possible moment. Many advanced templating features, such as variable text runaround of arbitrary shapes, or the flow of variable text between separate copyholes, are made more efficient by following this principle, which we call Just In Time Composition.
- The platform should integrate with existing workflows as easily as possible. The support of standard creative tools is obviously part of this but Anvil also has features which ease integration with user's dataflows and can produce PDF output when PPML is inappropriate
- Anvil aims to render VDP jobs at press print speed. Our in-house tests show that the Anvil platform is able to produce PPML jobs (from realistic templates) faster than the Indigo press can consume them

In the next sections we will describe the Anvil platform, its functional components and their relationship to the different stages of Variable Data Print workflows.

[1] SVG: Scalable Vector Graphics – a W3C specification for describing two-dimensional graphics and graphical applications in XML

Segmented Workflow Approach

A fundamental aspect of the Anvil approach is that the workflow is segmented[2] – different parts of the workflow take place on different software components, which can run on different computers. This approach contrasts with the monolithic workflow of, for instance, Yours Truly Designer[3] which bundles VDP template design, data manipulation/merge, composition and job output into a single application (actually, a Quark XTension). It is not that the monolithic approach is wrong, rather it does not suit all print service providers all the time. Specifically, the monolithic approach is more difficult to integrate with the kinds of complex, dispersed workflows frequently seen in larger PSPs. Furthermore, the monolithic approach forces one particular view on the VDP workflow. In YTD's case this is clearly the view from an extended static layout application. In the segmented workflow each tool can be given a UI and a function set most appropriate to the likely user. Finally, under the segmented workflow approach there is the possibility to replace one component with another of different or improved functionality, as long as the replacement is compatible with the underlying (standards-based) document format.

So what are the steps in this segmented workflow? In figure 1 below the dark blue blocks are the principal components of the Anvil Segmented workflow:

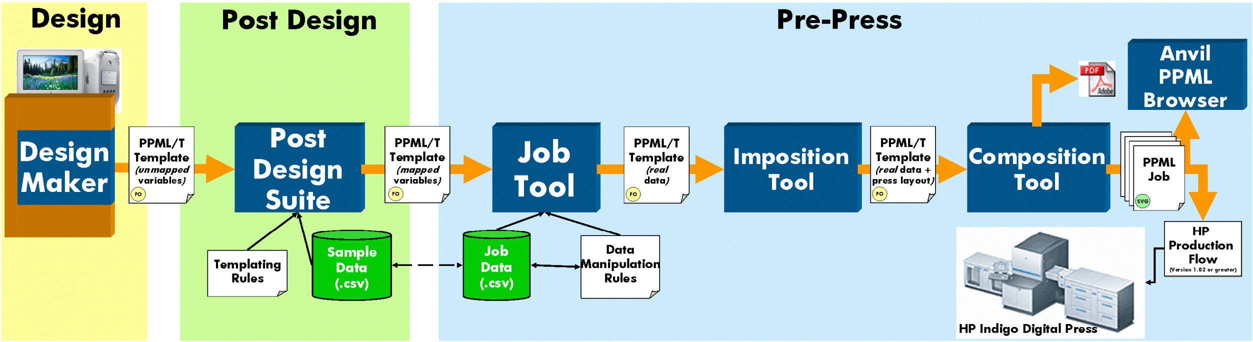

Figure 1: The Phases and Components of the Anvil Segmented Workflow

The diagram also shows how documents evolve as they pass through the workflow, and highlights the three distinct stages from creation to fulfillment of a VDP job. Note that the variable text is only composed into a press optimized job (with text in SNAP-SVG - see below) at the last moment. The Segmented Workflow increases inter-operability between the actors and entities within the publishing workflow by providing a set of components which support an underlying format called PPML Template. Defined in 2002 by PODi, PPMLT is, despite its name, a generic templating format, which defines and supports the three major components of a typical VDP job: Template, Data Mapper and Data. Let's analyze these three components in more detail.

The Template represents the outcome of the GA's design work. It holds the textual and graphic elements (both fixed and variable) defined by the GA along with the logic defining any intended customization. Since in a PPMLT Template there is no need to specify database structures, GAs can concentrate on the type of work

[2] The term Segmented Workflow, in the VDP context, originated with an earlier HP project called Real Time Targeted Marketing and much of the RTTM work has been inherited by Anvil.

[3] HP's current VDP plug-in for Quark XPress. Information available at: http://h30267.www3.hp.com/country/us/en/products/digital_press/workflow/yours_truly.html

that best suits them: design. At a later stage, the variable elements or "place-holders" can be linked with real database fields. This decoupling of the design from the database increases flexibility and robustness and allows changes to be made more readily to designs and database structures. It also ensures designers are not exposed to confidential data, which can be bound with the template later.

The Data Mapper contains the explicit reference between template variables and database fields, and provides a means to introduce business logic to drive the various template options. This business logic is usually represented as a set of rules that act on one or more database fields and trigger design options. Selection of particular content, or the inclusion/exclusion of template elements, and the printing or suppression of entire pages are examples of these options.

The Data represents the necessary and sufficient information distilled from the database to execute the job. Only database fields relevant to the job go in the PPMLT file.

Given this componentized Segmented Workflow, different scenarios for publishing workflows can be envisaged which enhance collaboration between the typical actors. Take, for example a customer such as a Marketing or Campaign Manager within an Enterprise, a GA who belongs to a separate design house, and a PSP responsible for the production stage. The scenario depicted in figure 2 illustrates the case where the business rules, the data preparation and the merging are executed on the customer premises, after the design stage. This clearly enhances customer security for the processing of confidential data.

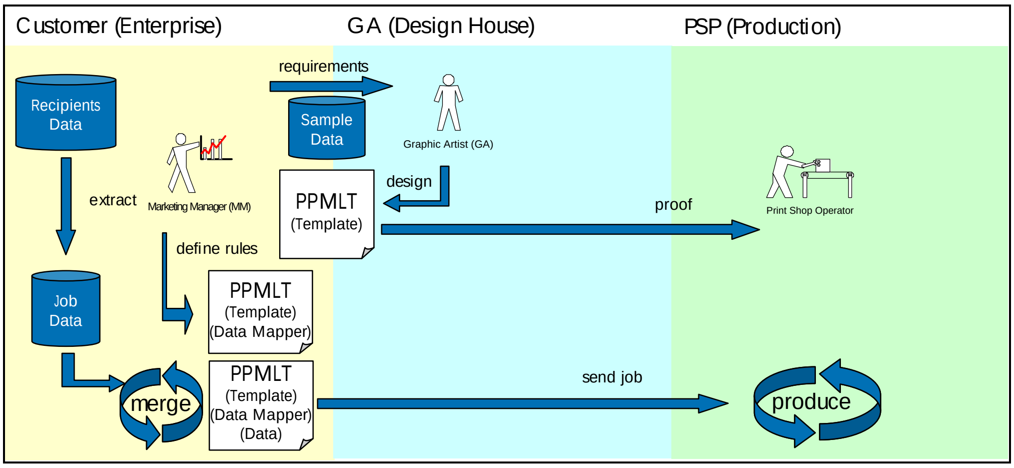

Figure 2: Scenario 1 - Customer (Enterprise) centric job data merging

The scenario depicted in figure 3 below illustrates a case where business rules are prepared on the customer's premises and where the data merging and production is done on the PSP premises.

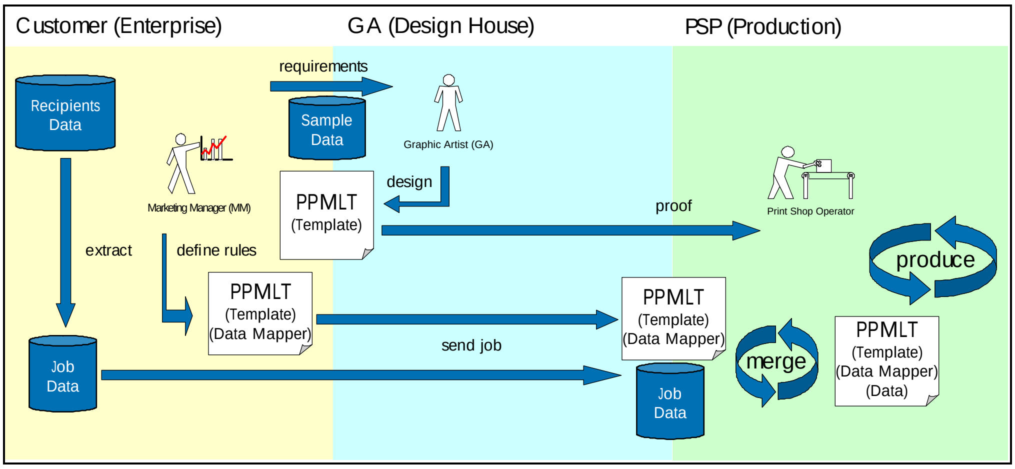

Figure 3: Scenario 2 - Print Shop centric job data merging

These scenarios highlight the flexibility gained by the adoption of a segmented workflow. They are illustrations rather than an exhaustive list and it is possible to envisage situations where the customer delegates all the work to the GA or the PSP or has it done entirely in-house.

Segmented Workflow Phases

It is convenient to describe the Segmented Workflow in terms of three phases: Design, Post Design and Pre-Press. This matches commonly accepted roles and phases in the conventional publishing sector and also locates each of the Anvil tools in a familiar context.

Design Phase

The creative phase is supported by Anvil Design Maker plug-ins for the leading static layout applications such as Quark XPress and Adobe InDesign. These plug-ins augment the static layout design tool, allowing GAs to add variability simply by selecting design elements (text or image copy-holes) and defining them to be variable. A key point is that Design Maker enables the creation of more sophisticated VDP template documents than, e.g., Yours Truly Designer. For example, variable text can flow between separate copyholes, or be flowed around irregular shapes, all with predictable rendering times during the job creation phase. A second key point is that Design Maker supports two kinds of variability: first there are simple variables (text or image) which will get their values from a field in a database when the job is created. The second type of variable is enumerated or complex, that is, the variable has a fixed set of possible values, specified at design time. These two types of variable match real-world VDP tasks very well: consider a personalized and customized print document destined for recipients in more than one country. Clearly, variable items such as the recipient's address will be different (personalized) for every record in the customer database and cannot possibly be specified at design time. On the other hand variable items such as an image of the recipient's national flag, or a block of known text in the recipient's language, can be specified in advance (customized). Design Maker allows the designer to create enumerated variables and adds a new control concept called a theme. A theme takes responsibility for a group of enumerated variables, whose job-time value can then be triggered off a single database field e.g. Nationality. As we will see below, this simplifies and adds power to rule-generated variability in VDP templates. Themes also reduce the need to introduce extra "control fields" into the database. Anvil authoring plug-ins, in-fact, can define two levels of customization: a content base customization, the theme, and a layer or layout base customization called a variant. Another term for this sophisticated capability is "variability within variability". Themes are capable of connecting two or more variable copy-holes under the same set of options, allowing the content to change according to high level customization dimensions, such as gender, nationality, age etc. Variants, on the other hand, can change the layout appearance, for instance introducing or removing a copy-hole. This allows a more fine-tuned level of customization or can be used to create a hierarchy of templates nested inside the same design.

As mentioned above, the Anvil layout tool plugins output the VDP template in PPML-T, expressing the main template format as PPML and the various variable elements as a combination of XSL-FO[4] and SVG. PPML-T is an open standard XML format which combines PPML with XSL-T. This means PPML-T documents can be executed, that is, the XSL-T can be run through an XSL-T processor to draw in data from a second XML file and combine it with the PPML part. However, Design Maker does not carry out this stage. Instead it leaves the template in a minimal state, decoupled from the real data (it is possible to include sample data for previewing and testing) but ready for connection in the next stage. One advantage of this decoupling is that it ensures the template is insulated from changes to the database. Furthermore, because it is decoupled from the database of personalized text and images the template emitted from Design Maker will be normally be small relative to an equivalent VDP job: Just the XML (PPML) defining the geometry of the layout, plus the fixed content, plus the XSL-T representing (a) the placeholders for the eventual insertion of the variable data and (b) the control structures representing the themes.

A vital point is that all text in copyholes that contain variables (we call such copyholes channels) is represented in XSL-FO which, although completely uncomposed (that is, not broken into lines or kerned) nevertheless captures all the essential font, style, and structural information to properly represent the designer's intention. This is a very important feature which allows Anvil to support advanced features in the template and yet still take advantage of Indigo's SNAP[5] accelerated text rendering at press-time.

Post-Design Phase

The post-design phase is covered by the Post-Design Suite - a Java GUI application architected as a skeleton application whose functionality is extended with PDS Plugins. The PDS opens the template and allows it to be examined in WYSIWYG fashion, highlighting variable portions of the layout and exposing the themes. The PDS enables Marketing Managers (MM), Document Engineers (DE) and IT Engineers to work on the template and the data. Several classes of operation can be done at this stage:

- template manipulation (re-size, merge and sub-template referencing)
- variable to data linking or mapping
- template theme and variant enactment using rules

The template manipulation stage enables the MM to combine different, separately generated templates by merging or page referencing. This mechanism enhances the capabilities of the template without making the design phase over-complex and indeed simplifies the GA's work. The PDS enables an entire page of another template to be embedded within the host template. When the merging operation is executed all the variables, themes and variants are merged and any resulting conflicts are solved. The page inclusion plug-in enables the MM or DE to include pages from other templates. The templates are completely independent from each

[4] eXtensible Stylesheet Language – Formatting Objects: a W3C specification for powerful yet flexible definition of document layout. Anvil uses XSL-FO to completely capture the content, structure and style of text elements.

[5] Swift Native Accelerated Printing – an Indigo technology that greatly improves rendering times for variable text.

other and any change will be reflected accordingly. Using this technology it is possible to split a complex publication into several templates at the page level and then re-combine them as variants, conditionally printing some pages according to the evaluation of the rules.

The re-sizing capability enables the MM to re-purpose the same design for different media sizes, i.e. from a A3 poster to an A4 leaflet or an A5 label. Note that this will be an exact scaling for text elements rather than a font scaling one: the scaled template will specify different font sizes. This offers a better way of overcoming the notorious Quark forty-inch/one-metre limitation on layout size[6]. PDS up-scaling will produce better text results than Quark up-scaling.

The Variable to Data Mapping allows the MM to wire up the template to the real database. This supports the decoupling of the design from the data, allowing the data structure to change over time without affecting the existing templates. This reduces the need to re-author the template (usually with GA involvement) just because field names or their order change (as they often do in the VDP world, where templates are re-used with fresh data as jobs recur and evolve over time).

The last capability allows the MM or IT person to write a set of rules that drive the customization aspect of the template. In this picture, the GA has already defined the suitable options that can be considered and has connected all the variable parts to it. This prevents an illegal or unplanned use of the customized portions inside the template. The rules are constructed via an easy-to-use wizard dialog and expressed in simple Boolean logic and IF-THEN-ELSE statements.

The PDS then writes the mapping and the rules (but not the database) into the PPML-T template. At this point the template has grown in size by a small amount – the size of the data-mapper element and the rules. We envisage extending the PDS with more plugins, such as data validation tools.

Pre-Press Phase

The pre-press phase prepares all the constituents of the job into a suitable form for consumption by an Indigo Digital press or other device, such as HP DesignJet or HP LaserJet.

During this phase of the segmented workflow several operations are carried out:

- the data is merged with the template producing a job
- the template or the job is imposed using imposition instructions
- the job is composed and prepared for the printing process

The Job Tool is used to carry out the data merge. Also a Java GUI application, the Job Tool allows the user to select a standard CSV or XML data file and import its variable text and image references (in Anvil, images are only referenced from the template, not included within it) into the template. The Job Tool can now either export the PPML-T with a new section containing the data, or it can export the template as a fulfilled (but uncomposed) PPML file. The former is ready for passing to the next step of the workflow; the latter may be examined in the PPML Browser (described below) to check that the job is on course. The PPML-T file is no longer purely a template (but is not yet a press-ready job either) and has now enlarged by an amount equal to the size of the imported records (plus a small overhead due to it being held in XML). The exported PPML file

[6] Quark sets an upper size limit of one metre on layouts. Designers commonly overcome this by outputting with a scale factor of more than 100%.

will, of course, be larger due to PPML repetitions but with effective use of a table of re-usable objects (text or image items that are used more than once) an optimal file can be produced.

The Imposition Tool performs the next step in the Anvil workflow. A Windows native executable (architected as a DLL and a .Net GUI front-end), the imposition tool opens a PPML or PPML-T file and allows the user to impose the layouts it contains onto the target substrate size. The Imposition Tool is essentially a standalone clone of the YTD imposition capability: it offers a choice of several preset substrate sizes or a custom size; and it comes with 16 preset imposition templates, such as 3 Up Duplex etc, as well as allowing user created impositions. Page orientation and page numbering can be applied at this point. As with YTD, automatic step and repeat can be applied to efficiently fill impositions with data-driven document instances at press-time; imposition marks can be added, bleed and gutter parameters can be set, but more advanced imposition techniques, such as creep or shingling are not yet supported. The Imposition Tool writes all these parameters in standard PPML into the document. Although the Anvil Imposition Tool offers little functional advantage over YTD it does have practical advantages: following the segmented workflow philosophy it can apply imposition to either a template or a job and it doesn't require committing to the imposition at the design stage. So, if the imposition requirements for the print job change, these can be modified without affecting the entire workflow. This flexibility is also important when the template has a variable number of pages (usually from a maximal set), meaning that the imposition can only be carried out when the overall number of pages has been defined (i.e. at job stage). On the other hand, in the case where the number of pages is always the same for each record the imposition can be defined at the template stage.

The Composition Tool, another standalone Java GUI application, carries out the final stage in preparing a job for the press. At this stage all the variable data has been merged inside the corresponding copy-holes and the layout composition takes place - transforming the job into the final format that the press or printer can process. The composition stage is completely independent from the authoring environments and is completely controlled by Anvil, thus ensuring predictability of results and automation of operations to detect and prevent situations like overflow or underflow. Because all the uncomposed text elements are expressed using the XSL-FO notation, Anvil has developed an augmented version of Apache's FOP (Formatting Object Processor) called AS-FOP[7]. which can render this to a press or printer optimized format. There are several advantages in controlling the composition stage:

- AS-FOP can compose text into press optimized SNAP
- AS-FOP has a feedback API which flags text overflow and underflow situations allowing the Composition Tool to apply fallback actions such as font resizing
- We gain the possibility of generating formats for different devices, such as PPML for Indigo Press, PDF for HP LaserJet and HPGL for HP DesignJet.

The composition tool uses full font metrics, kerning and hyphenation descriptions to give a composition that has similar quality to professional typesetting tools, like native Quark XPress and Adobe InDesign static layouts.

Finally, the Anvil Segmented workflow provides the PPML Browser Tool. This is another Java standalone GUI application which allows the press-ready PPML to be examined WYSIWYG-style before committing to a press run. The PPML Browser can validate the PPML job against standard XML Schemas, or user supplied custom schemas, ensuring integrity of the job. It also analyzes the PPML file's compliance with HP Indigo PPML support and can also display various statistics – size, list of re-usable objects etc - of the PPML file. This tool

[7] AS-FOP: Arbitrary Shape FOP. Anvil's version of FOP can flow variable text into regions of arbitrary shape and render the results efficiently.

can be used as a soft-proofing application to evaluate how the templating rules behave and explore corner cases with specific variable data instances.

Conclusions

The segmented workflow approach and the Anvil toolset provide substantial advantages compared to current VDP workflow solutions. Explicit separation of concerns and a more flexible interaction between the different parties involved are key aspects. The Anvil plugins for layout tools allow designers to create more sophisticated VDP templates than competing solutions, and lead to simplified real-world VDP solutions. In addition to the toolset, the underlying VDP document format – a combination of PPMLT and XSL-FO - ensures complete decoupling of the authoring environment from the later stages, reducing GA costs. The late binding stage opens the possibility of applying text adjustment procedures to automatically recover from some composition errors, increasing the accuracy and predictability of results. Finally, the component-oriented Anvil solution allows for easier integration with the wide range of user installations to be found in this growing sector.

Acknowledgments

The Anvil Team is: Guilherme Bedin, Thiago Boi, Alexis Cabeda, Fabio Giannetti, Douglas Michaelsen, Fabio Moreira, Andrei Nunes, Anderson Oliveira, Ricardo Pianta, Hugo Rodriguez, Royston Sellman, Tony Wiley.

For More Information, go to www.hp.com/communications
Document location: (i.e. http://someserver.hp.com/file.doc)