

# FLM: Six Steps For Effective SAP Interactive Forms

Chris Scott, Arch Consulting



*SAP Interactive Forms by Adobe can help to liberate business processes that are stymied by paper or forms requiring manual entry into SAP.*

*Using Forms Lifecycle Manager (FLM) functionality, businesses can rapidly deploy new, integrated processes and rapidly reap enormous business benefit.*

*There are six key steps to ensure that the introduction of forms-based processes is both smooth and successful.*

## STEP ONE: Start with the Business Process.

Spend time to design the new business process properly. This may sound obvious, but in practice, business managers can find it too tempting to use new technology to replicate an existing, old business process.

With the introduction of any new technology the business process will undoubtedly change to some degree. For example, it is likely that the introduction of a SAP Interactive Form will remove the need for double-entry at some point, or the need for discrete solutions involving paper forms/Microsoft Word/Excel. Alternatively the electronic form might enable a whole new business process.

It is easy to precisely replicate an existing Microsoft Excel or paper form based on the argument that this will maximise user adoption. However, this is a lazy approach, and you are likely to be missing opportunities to streamline the business process *and* enhance the user experience.

**Choose forms for the best-fit processes.** For example, SAP Interactive Forms solutions best suit processes involving infrequent SAP users, so forms users generally may not have access to the SAP system. Forms solutions work well for collaborative processes, whereby several users participate in a workflow before the final SAP document posting.

One of the earliest and most important decisions you need to make is whether to design an 'on-line' process, through which forms are rendered dynamically through a portal or other application request, or on 'off-line' process, through which forms are delivered to users using e-mail.

Select *on-line* or *off-line* form scenario (or even a mix) to best fit the new business process. Factors you need to consider include:

- Access. You may wish to provide employees with access to your forms portal, but not external partners. If an external partner is involved in an on-line process then your portal must be available outside your firewall.
- Security. You may prefer to control all forms access using SAP authorisations, which suggests an on-line approach. Alternatively you may prefer to use PDF documents which travel as e-mail attachments, with your business data safely encrypted within the content of the PDF.
- Traffic. You may prefer to e-mail out a form rather than require many thousands of users to access your forms portal at the time and day for recurring forms processes like timesheets.
- Business process. For each particular process do you want to require your users to log in to a portal, or to respond to an e-mail?

## STEP TWO: Don't ask too much of a form process.

A form should be designed for a single business process, not several. General-purpose forms are harder to implement and harder to maintain. If a single form needs to look different or behave differently for a particular business unit or business partner, then create a separate version of the form for that purpose.

We worked with a local authority in the UK that was using more than twenty different timesheets across its business units. The timesheets required particular fields or particular SAP postings for particular business units. In this scenario the challenge is to make future form maintenance as easy as possible; which meant that the ideal number of electronic forms was more than one but less than twenty, the specifics of the business unit's requirements enabling versions of the same form to be rolled out across multiple business units.

**Don't try to use forms to front-end SAP.** Interactive forms don't replace the SAPGUI or the rich functionality of enterprise portals that can link to multiple enterprise applications. A form could certainly trigger the creation of a SAP document like a sales order or purchase order, but would typically contain only a subset of the available fields.

If your business requirement is for a functionally-rich employee self-service portal, then using SAP ESS is probably a better approach than building the same functionality using interactive forms!

### STEP THREE: Design an irresistible user experience.

**Only display fields that each user needs.** Throughout a form's lifecycle, different form owners will need access to different fields. Hide the information that they don't need to give them a high quality experience. FLM provides functionality to dynamically hide or highlight different form fields at different points in the form's lifecycle.

**Provide meaningful and useful validation messages.** Ensure that each form field is validated such that user's are prevented for submitting incorrect data wherever possible. Some validation can occur within the form, such as ensuring quantity fields have a positive value. Other validation occurs at the server-side when the user attempts to submit the document. FLM provides protection for users from SAP messages by de-coupling the form submission from the final SAP update.

**Dynamically derive everything you can.** Do not require the user to enter information if the data can be derived by the system. Derive as much as possible from the user id and use user parameters if necessary. If fields can be derived rather than entered based on other form data then derive them at the server-side after form submission or during the final posting. Consider:

- [a] Whether you can use an existing FLM Posting Adapter by deriving data earlier in the lifecycle, and
- [b] Whether derived data is required during the lifecycle of the form. In this case it should be derived at the point of form render or submission using one of the FLM business services.

**Make use of PDF accessibility functionality.** The accessibility of PDFs can be greatly improved by adding 'tags' to the form. These contain information about the structure of the document such as header locations, lists, tables, hyperlinks and alternative text descriptions for graphics. This allows users of assistive technologies such as screen readers to understand the document more easily and navigate within it.

### STEP FOUR: Store your business logic within SAP, not within the form.

**Don't store Business Logic within the form template.** Accompanying each form template is a set of business logic, such as drop-down list possible entries, locked or hidden fields and field validation rules. If such logic is stored within the form template using hard-coding or JavaScript, then form maintenance becomes extremely difficult in a changing business environment, particularly when a large number of form templates are being managed.

FLM permits you to centrally store all business logic within SAP configuration tables and simple user-exits, available through the SAP IMG. Using this logic, FLM automatically re-renders the form at each stage of its lifecycle. Fields can be hidden, protected or highlighted throughout the process, form data can be pre-populated, derived or validated, or a different form template could be used.

Business services that should run before a form is rendered include:

- *Derivation of form template based on form type, version and language.* It is important to be able to manage multiple versions of a form type in order to manage business process change.
- *Pre-population of form fields.* Form fields that can be pre-populated based on SAP user data, organisational structure or document data should be derived before the initial form render.
- *Pre-population of from drop-down lists.* The contents of drop-down lists should be dynamically generated based on SAP tables rather than being hard-coded or relying on web services which may slow down the form response time or not work in the off-line scenario.

Business services that should run on form submission include:

- *Field validation.* Validation that relies on SAP data or represents business logic rather than form logic should be handled at the point of form submission in order for the user to receive an instant validation message and for them to be given the opportunity to correct their mistake before the form progresses further.
- *Field derivation and substitution.* Form data that is derived based on the contents of other data fields should be derived after form submission unless it is immediately necessary to the form user, in which case a 'data check' update or a web service call is required, triggered by a form action such as 'field exit'.
- *Derivation of form routing logic such as the new form owner and new status.* Each time a form is submitted we should derive what happens next to the form within its lifecycle; be that an escalation to a new owner, or a SAP posting.

**Store Form/Data Logic within the form template.** In addition to these 'server-side' business services, it is possible to code 'client-side' form logic against each field and sub-form using JavaScript. This is useful for dynamic calculations like 'total' fields and simple checks such as a numeric field value being positive or a date being in the past. For this kind of data check it is not necessary to wait for a server-side validation.

#### STEP FIVE: Utilise event-driven, data-specific form presentation logic.

**Use client-side scripting to control the form look and feel based on the data entered by the user.** This can greatly assist in reducing user entry error, and can enhance the user experience. For example, if a user selects a particular checkbox to enter more data then a subsequent set of data can be dynamically displayed or highlighted based on the value of that checkbox.

In this way the properties of every field or sub-form can be dynamically updated to be:

- Visible or hidden
- Highlighted or 'greyed out'
- Available for entry or read-only

At the server-side FLM enables these properties to be changed on form render based on the status of the form, which represents the point in the form's lifecycle. At the client-side, the addition of JavaScript further enables these properties to be maintained based on form events such as field exit or calculate - and based upon the current form data.

#### STEP SIX: Design the solution in the right order

**Build a prototype quickly.** Where possible replicate the same look and feel of those existing paper or non-integrated forms that the Interactive Form will replace. This ensures high user adoption and enables you to build a form prototype quickly if internal selling of the project is required.

**Design the process first, then the SAP posting, and then the form.** We need to understand the requirements of the data schema for the full form lifecycle before we can design the layout of the form. The process, the form and the final posting all impact the data schema that is required for the form. The most efficient process of forms design should be:

1. *Form 'look and feel' prototype.* This can take less than one day, and immediately give the requesting party something 'real' to play with; only when they try to fill in a form do all the business requirements begin to emerge.
2. *End-to-end process design.* Document all the business logic for form routing, especially considering exception processes such as form rejection and cancellation. Understand what data can be automatically derived and what information is necessary to be on the form at each stage in its lifecycle.
3. *SAP Posting Design.* Identify an existing BAPI [or design a new function module] for the final SAP posting and understand what data is required in order to successfully call the BAPI. Decide what data for the posting needs to be stored within the form versus being derived before the BAPI call.
4. *Logical Form Design.* In FLM use the data designer within the form wizard to design the data schema, which is the structure into which all the form data will be stored throughout the form's lifetime and whenever the form is rendered.

5. *Physical Form Design.* Based on feedback from the initial prototype, and based on the schema based on the process and the posting, a final form layout should be developed - and this can also take less than a day. Now add the client-side script for event and data-driven presentation logic.
6. *Add the business logic.* Develop business services at the point of form render and submission and configure field attributes to control the form look and feel dynamically based on the form status.

Until recently this whole process would take weeks or months, but with FLM can be accomplished in just a few days.

### Summary

New business processes using SAP Interactive Forms can enable real business efficiency savings and can be developed extremely rapidly using Forms Lifecycle Manager.

Following the 6 guidelines described above can result in highly successful projects and open up the SAP enterprise system to processes and to new users.

---

#### ABOUT FLM

FLM exposes SAP Interactive Forms functionality through SAP configuration and simple ABAP user-exits.

This removes the need for Netweaver Development Studio, Java Webdynpro, Java, JavaScript and complex ABAP coding - all of which are normally required when building a SAP Interactive Forms solution.

FLM also provides run-time framework for forms management, including form output, routing, posting and administration components.

For more information, see [www.arch.co.uk](http://www.arch.co.uk)

---