HOW TO DESIGN AN END-USER INTERFACE
WITH EMC DOCUMENTUM xCP
Best Practices Planning

Abstract
This white paper discusses the EMC® Documentum® xCelerated Composition Platform (xCP) in the context of interface design for automated, case-based business solutions and applications.

February 2011
# Table of Contents

Executive summary .............................................................................................................. 4
Introduction .......................................................................................................................... 4
  Audience ............................................................................................................................ 5
Documentum xCP overview ................................................................................................ 5
  Enterprise content management ....................................................................................... 5
  Service-oriented architecture ............................................................................................ 5
  Content management and repository services .................................................................. 5
  Business process management ......................................................................................... 5
Interface design with Documentum xCP .............................................................................. 6
  Designing a task view ....................................................................................................... 6
  Display a document package inline .................................................................................. 7
  Display a folder package inline ......................................................................................... 7
  Display multiple packages inline .................................................................................... 8
  Custom error messages ..................................................................................................... 9
  Task template buttons ...................................................................................................... 9
  Interface design and performance .................................................................................... 10
  Data adapters .................................................................................................................. 10
Conclusion .......................................................................................................................... 11
Executive summary

EMC® Documentum® xCelerated Composition Platform (xCP) is a single, integrated platform that uses configuration rather than coding to enable development of comprehensive, case-based business applications and solutions. As a composition platform, Documentum xCP provides reusable components, model-based configuration, a configurable user interface, dashboard tools, and reference applications to enable developers and IT architects to compose and maintain applications using graphical tools.

Introduction

Managing business processes via automated, case-based solutions can improve process efficiency and effectiveness in a variety of ways. These solutions can:

- Minimize the reliance on time-consuming, error-prone, paper-based systems to compile and manage process data
- Eliminate the need to locate and physically transport information stored in file cabinets
- Enable continuous monitoring and control of processes through realtime dashboards
- Allow process participants to collaborate through discussion threads
- Provide security and information rights management for confidential information
- Employ a secure repository to meet government requirements for compliant records management

But if the solution interface is not designed correctly, these expected benefits will go unrealized. Instead, the solution will simply become “shelfware.”

User interface design—more than functionality—is often the difference between the success or failure of a solution. If end users find a solution cumbersome and difficult to understand, it is likely doomed to failure. A good interface enables users to perform tasks quickly and efficiently, without sacrificing functionality or flexibility.

This white paper examines interface design with the tools provided by the EMC Documentum xCelerated Composition Platform (xCP). Documentum xCP includes a set of fully integrated technologies, with field-tested development and deployment tools. xCP also provides a set of xCelerators that enhance the delivery experience and speed deployment of applications. These xCelerators include process templates and blueprints, data structures, predefined integration points, sample dashboards and reports, Express Installers, best practice guides, and sample applications.

For more information on interface design, please refer to our comprehensive, 160-page EMC Documentum xCP Best Practices Guide. You can also visit the EMC Developer Network at https://community.emc.com/community/edn for additional resources such as e-learning modules and tutorials, web content, white papers and...
articles, technical documentation, podcasts, videos and webinars, software downloads, code samples, news, and topical reviews. For a best practices overview of Documentum xCP, please refer to our companion white paper How to Design and Implement Business Processes with EMC Documentum xCP — Best Practices Planning.

**Audience**

This white paper is primarily intended for user interface designers who are part of an xCP solution development team.

**Documentum xCP overview**

Documentum xCP is a single, integrated platform that leverages Documentum enterprise content management (ECM) and business process management (BPM) products to develop comprehensive, case-based business solutions.

**Enterprise content management**

The Documentum ECM platform provides a unified environment for capturing, storing, accessing, organizing, controlling, retrieving, delivering, and archiving any type of unstructured information across an extended enterprise. With support for Web 2.0 information requirements and high volume transaction and archive environments, the Documentum content management platform supports global enterprise deployments.

**Service-oriented architecture**

Service-oriented architecture (SOA) looks at IT assets as service components. The SOA approach to building business applications is based on creating standalone, task-specific reusable software components that function and are made available as services.

**Content management and repository services**

The Documentum Content Server and repository manage information and content with security, scalability, reliability, and a common set of policies for all content assets across an organization. The Documentum repository also manages the application definition artifacts, configurations, and supporting models.

**Business process management**

Documentum xCP provides BPM capabilities that integrate fully with the ECM platform. Using a model-driven approach combined with services orchestration capabilities, Documentum xCP provides the ability to configure applications without coding. Documentum xCP also provides a complete range of modeling tools for processes, tasks, data, content, presentation, and business logic. To modify a solution, developers simply change the model, which declares what the application needs to do, instead of writing code that specifies how it’s done. This provides an agile framework for building, maintaining, and improving business solutions.
platform’s service-oriented architecture also enables easy integration with existing line-of-business (LOB) applications and external systems to extend the value to those investments.

**Interface design with Documentum xCP**

User interface design takes place in TaskSpace, with the templates provided in Forms Builder. A good UI should be as intuitive as possible, with a common look and feel across all forms.

Before work begins, it is a good idea to confer with the client and prepare a detailed design document that addresses as many UI issues as possible in advance. This makes creating the forms a much more straightforward process. A visual style guide is another valuable document to create with input from the client.

Clients should review forms often during the design phase. They are the end users, so their feedback is critical. Nevertheless, sometimes what the client wants does not contribute to good design. For example, just because it is possible to search on 20 attributes doesn’t mean that it’s a good idea to include that functionality in the design.

To ensure a successful UI, consider the following tips and recommendations:

- Calculate how many task templates are needed. For example, each step or activity in a process could have its own template. This approach can also improve solution performance.
- Make a fully functional prototype to help ensure that the UI design is user-friendly. Don’t begin development until the client signs off on the prototype.
- Keep in mind that due to coding complexity and customization, development time for the complete interface will be greater than for the prototype. Also, prototype testing will not necessarily identify performance issues.
- When designing templates, it is sometimes faster and easier to delete a failed template design and start over.
- Try to design forms that do not require scrolling. This may mean using more tabs in TaskSpace.
- While making changes to a template, frequently verify how the changes display in the preview pane. When changing and adding controls and moving items on the canvas, spacing and padding will appear differently in the design pane than they will in the preview pane or in production. This is especially true when editing templates using drag and drop. To ensure correct alignment, especially for labels, use the settings on the style tab.

**Designing a task view**

Since processes and tasks are so intricately linked, a well-designed task view can contribute substantially to user productivity with a case-based application. To design
a task view, use the task template creation wizard, selecting a process, process variables, and the default controls. Once the task view is created, custom logic, such as adapters or an invoke button, can be added. With the default task template, users can view attachments or packages in the TaskSpace preview pane. There are also several custom options for designing a task view.

**Display a document package inline**

A task view can display a specific document package inline using the task template with an embedded form control. When selecting data binding for the embedded form, be sure to bind to the valid object path: /Activity/.../<package name>/DocumentId. Forms can also be bound to a process variable if the variable value is an object ID.

Next, select the appropriate form template to show the package content, which is either a document view template or an electronic form. Within this task view, it is possible to preview a document and modify document data while processing a task. A single task view can accommodate multiple embedded forms, but this can make the interface chaotic and busy-looking.

![Figure 1. Task template with embedded document view](image)

**Display a folder package inline**

Using the task template with a folder view control, a task view can display a specific folder package inline.

Select a folder package for data binding and configure the folder view display. This option is ideal for case-based applications since it allows a folder to be treated as a
case and enables the user to preview a document while processing a task. Space constraints will limit the amount of information that can be shown.

Figure 2. Task template with folder view control

Display multiple packages inline

To display multiple packages inline, use the task template with a folder view control and an attachment list control. This option will accommodate multiple types of packages such as documents and folders but requires a careful use of space to maintain an uncluttered interface.

A folder view can display a folder object from the attachment list. Through the preview pane, users can view an object from the folder view or a document from the attachment list. When viewing a document from the attachment list, the folder view is empty.
Custom error messages

The custom error message is an important element of interface design since it provides useful feedback to the user when entering data. Error messages are created in Forms Builder by the template designer and are entered in the error message field for each control on the Display tab in the Properties pane.

Custom error messages should give the user specific information, describing the correct value to enter. Simply stating that an entered value is invalid does not provide much guidance. Phrase error messages in a positive manner and provide an example of an acceptable value. For controls that use specific masks, such as dates, Social Security numbers, and phone numbers, provide a default value. This way, a user has an example to follow before entering a new value.

If a control does not have a custom error message defined in the template, an incorrect value will trigger an error message. Default error messages only describe the data constraints for the control. For example, a default error message for an invalid data type is “Value entered is not valid for the integer data type.” Likewise, a default message for an invalid value for a specific data constraint is “Enter a valid value equal to or greater than %% (where %% = the defined minimum value).” These messages are not nearly as helpful as messages created specifically for a template, tailored to the template’s data requirement.

Task template buttons

Since task templates are complex and closely connected to processes, problems can arise when underlying processes change. The next sections provide guidance on task template buttons that can be affected by changes to the process or data model.
Adding task buttons

When a task template is created, Forms Builder automatically adds the appropriate buttons based on whether:

- There are one or more forward tasks
- Forwarding is conditional or manual
- There are reject paths

Nevertheless if process flows are removed or added when a process is modified, these buttons remain static and will not necessarily reflect the new process flows. During testing, it may not be possible to finish or forward a task. To avoid this problem and save time during development, simply add all the task buttons that may be needed. TaskSpace will automatically show and hide buttons as appropriate.

Name buttons with the same name as the actual activity. For example, if it is a reject button, label it **Reject**. The button may have a custom display label, but its name should reflect the activity being performed.

Finish button malfunction

If the underlying data model (such as a structured data type definition) is changed, update the task template to reflect this change. Otherwise the finish button may fail to work. Follow these steps:

1. Uninstall the task template.
2. Save the task template.
3. Install the task template again to refresh the template’s data model.

Interface design and performance

When working with form templates, performance and usability should always take precedence over appearance and special effects. With that in mind, here are some general performance guidelines for working with forms:

- Reduce complexity in form templates with fewer fields and fewer adapters.
- On a task list template, keep the search parameters simple.
- While using multiple task templates for different activities may improve performance, it also increases maintenance. A better approach may be to use one task template with conditional display settings.
- Keep everything optimized for the chosen database.

The next section goes into greater detail on the performance impact of data adapters.

Data adapters

Data adapters and rules all affect the length of time it takes to open a form. Using too many data source adapters can impede application performance. It is best to use one data adapter to populate multiple fields by sourcing from one database table with
multiple columns. This way, the adapter runs only once. Simple, lightweight adapters with short execution times exert the least drag on performance. Keep the following recommendations in mind.

- Do not perform high latency calls within data source adapters, such as long-running queries or slow web service invocations. These calls can greatly impede performance of the UI.
- Make sure one adapter works before creating another.
- When using a data source adapter, ensure that the input is required and that the output is repeating.
- When creating a custom data source adapter, create it as a class first. This makes it easier to validate and debug. Then it can be converted to a service-based business object (SBO) if desired.

**Conclusion**

The EMC Documentum xCelerated Composition Platform enables developers to leverage a case management model to build process-centric or case-based solutions for any line of business, with minimal custom coding. The first step toward a successful implementation is designing an easy-to-use interface that addresses end-user requirements.

Documentum xCP is the ideal development environment for building solutions and solution interfaces that increase process efficiency. To learn more about this new standard in cost-effective application development, visit us on EMC.com at [http://www.emc.com/products/detail/software/documentum-xcp.htm](http://www.emc.com/products/detail/software/documentum-xcp.htm).