K2® LEARNING

K2 Business Analysis

Course Handbook

Version: Version 3 Revision 1

Date: January 2016

"Terms Of Use" on page 278

Table of Contents

K2 Business Analysis Workshop: Course Overview

Course Overview

Estimated Duration: 12 hours
Version: Version 3 Revision 1(What's new)
Publish Date: January 2016
Based on product version: 4.7 (Later versions of K2 may have additional features or functionality not covered in this learning module)

Course Description

The “K2 Business Analysis Workshop” will give participants an understanding of how business opportunities and requirements are evaluated for implementation as K2 Business Applications. Intended for Business Analysts and Business Leaders, this two-day workshop-style course includes discussions and practical exercises that will help participants learn how to analyze, evaluate and gather requirements for Business Applications that leverage the power of K2.

Who should take this course

The K2 Business Analysis Workshop is intended for Business leaders (such as IT managers and executive sponsors), Business Analysts and Project Roles such as Project Managers, Development Team Leaders and Enterprise Architects. Day 1 is particularly suited to anyone who just wants to get a business-oriented overview of the K2 platform. Day 2 is focused more on the Business Analysis side of K2 Applications. There is no need to attend day 2 if the participant will not be involved in the Business Analysis process in any way.

This course is a 100-level course and participants do not need any K2 knowledge or experience prior to starting this course (eee additional requirements in the Course Prerequisites section). After completing this course, participants can continue on to other K2 training courses if they wish to learn more about a specific component of the K2 platform.

What you’ll learn

During this training event, participants will learn:

- A high-level overview of the K2 platform and what the platform means for business
- How K2 works with Business Process Management principles (BPM) and why BPM is valuable to organizations
- How to effectively implement BPM in an organization using tools like K2
- How the platform components work together to build business applications
- How to build a simple workflow application in K2
- How to qualify, analyze, discover and document functional requirements for K2 business applications
- How to approach requirements gathering for K2 applications in terms of Workflows, Forms, Reports and Data
- How to build custom reports in K2 (optional module)

Note
This workshop is focused on Business Analysis for K2 Applications. While we will cover some fundamental aspects of building applications with K2 and participants may gain some hands-on experience by building a simple application in K2, users who will be building production-quality Applications using the K2 platform should attend the courses offered by K2 that cover building of applications with K2, such as the K2 Blackpearl Core and K2 smartforms builder training courses.
Since this training event is structured as an interactive workshop, we encourage participation in the discussions and white boarding sessions to get the most benefit out of this event

Requirements

To get most out of this training course, it is highly recommended that the participants have the following skills and proficiency levels.

Technology/Skill	Proficiency	Notes and examples
Understanding of business analysis in Enterprises	Basic	■ A basic understanding of the theory and practices of Business Analysis
Understanding of technical aspects of applications	Basic	■ A basic understanding of what SQL databases and web services are ■ A basic understanding of user interface terms like controls, events and web pages

Course Learning Materials

This online site contains the course materials for this training course, including the hands-on exercise step-by-step guides. If required, the course materials are also available as downloadable files.

Item	Format	Notes
Course Handbook	PDF	The student guide (slide handouts) for this course, excluding the hands-on exercises
Course Exercises Guide	PDF	The step-by-step exercises for this course, excluding the slide handouts

Environment for practical exercises

This course includes exercises where participants will build K2 applications and work with K2 in a live K2 environment. For training events led by a K2 instructor or purchased with K2 credits, a K2 Virtual Machine will be provided in either local (virtual machine) or hosted (cloud) modes. Depending on the course logistics and delivery mechanism, K2 may provision the hosted environments ahead of time or participants will use on-demand provisioning to set up their virtual environments. Please refer to the K2 Knowledge Base article KB001397 for more information about the different Virtual Machine options and the prerequisites for each approach.

For hosted (cloud) virtual machines, participants should verify that they are able to access the virtual machine as described in the Knowledge Base article KB001279 prior to the training event starting, to allow sufficient time to troubleshoot connectivity issues.

For self-directed training where K2 does not provide a cloud-hosted virtual machine, participants may download a Virtual Machine image and use virtualization software to run the machine. This procedure is described further in the Knowledge Base article KB001613. Alternatively, the hands-on exercises in this training course can be repeated in any environment with K2 4.6.9 or later installed, provided that the participant has sufficient rights to deploy K2 applications to their K2 environment. When using your own environment to complete the exercises, some screenshots may be different (for example when searching for users or environment-specific settings shown in screenshots), but fundamentally the exercises can be rebuilt in any K2 environment.

The practical exercises for this training course can also be completed in any K2 environment, as long as you have the necessary permissions to deploy application elements and administer the K2 environment.

Course Content

This course will take approximately 12 hours to complete (excluding breaks and depending on the additional exercises selected). The course is divided into five modules:

100.ABZ - Introduction to K2

The *100.ABZ: Introduction to K2* training module is a high-level introduction to K2 and covers the following:

- Explains how and why K2 is used by enterprises
- Overview of the K2 platform components and products
- Explains the core elements of K2 applications (Data, Forms, Workflows, Reports)
- Describes the tools used to Build, Deploy, Use, Report and Administer applications in K2

This is a very high-level introduction and overview of K2, and there are no practical hands-on exercises or demos in this module. However, we will be describing some core components and uses of the K2 platform that help to set the scene for the next series of learning modules in this course.

100.VCP - K2 in the Enterprise

This learning module introduces the K2 platform and explains how the platform is used to address business requirements. It also covers the K2 platform components on a high level with practical examples of how these components are used in a sample K2 application.

By the end of this module, you should be able to:

- Explain how the K2 platform addresses business requirements and describe K2 platform components in general
- Define BPA, BPM, and BPMS
- Explain why organizations use BPM and BPMS and explain why organizations use K2 for BPM and BPMS
- Explain what SmartObjects are and what they do
- Identify the five phases of BPM maturity development in organizations and identify the six stages in the BPM life-cycle

100.BHX - Introduction to K2 Applications with K2 Designer

The *100.BHX: Introduction to K2 Applications with K2 Designer* training module explains how to build K2 applications in terms of Data, Forms and Workflows. In this module, we will use K2 Designer to build an application with Data (SmartObjects), Forms (SmartForms) and Workflow.

This module covers the following concepts:

- Using K2 Designer to build simple applications
- How Data, Forms, and Workflows are used to build an application
- Integrating with external systems with SmartObjects
- Integrating with external systems with Workflow wizards
- Using SmartObjects in Forms and Workflows
- Workflow concepts: escalations, task slots, workflow patterns

Although this module focuses on the web-based K2 Designer which may or may not be used by all organizations, we will be explaining some fundamental concepts about the elements that make up an application, and the knowledge gained in this module will set the scene for other leaning modules that delve further into Workflow and SmartObjects. Therefore, even if you do not intend to use K2 Designer in your organization, we recommend that you complete this learning module to build up your initial knowledge of K2 applications.

200.BEL - Business Analysis for K2 Projects

This learning module describes how to qualify, analyze, discover and document business requirements for K2 Applications. At the end of this module, participants should have a clear understanding of the typical life-cycle of a K2 project and how to perform the Initial Discovery, Analysis and Functional Design stages of a typical K2 project life-cycle, optionally using document templates provided by K2. We will discuss how to qualify requirements through the initial discovery and analysis for K2 projects and how to discover and model the components of K2 applications: Forms, Workflows, Reports, and Data.

Note that this module is not intended to cover the technical design, implementation, or development of applications with K2. Those tasks are typically performed by more technical roles like designers and developers and is covered in other K2 training courses.

100.CWL - Reporting in K2 (Optional Module)

The *100.CWL: Reporting in K2* training module explains how to use the available standard and custom reporting in K2 to report on workflows.

This module covers the following concepts:

- The standard reports that are available in K2 and where you find and run those reports
- Using the K2 View Flow report
- Creating custom reports in K2 Workspace and K2 smartforms
- Creating custom reports with other third-party reporting tools

This module is optional and can be completed in our own time if there is not enough time in the training session to complete this module. If your job responsibilities include Business Intelligence or reporting, you should complete this module, otherwise it is not required.

Additional content and resources

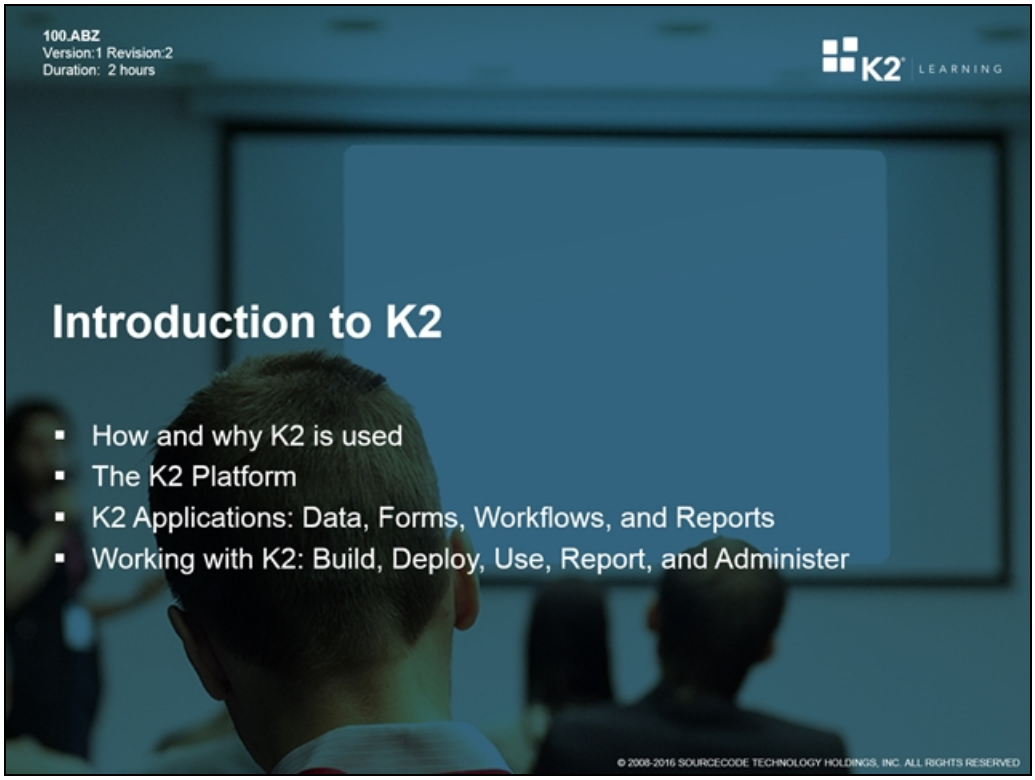
Beyond this course, the following resources and content are also available to users of K2 blackpearl.

Content/Resource	Overview and notes
K2 blackpearl Product Documentation	The official product documentation for K2 blackpearl
K2 Community Site	A community of K2 professionals featuring forums, blogs, samples and other resources
K2 Knowledge Center	K2's Knowledge Base
K2 Product Support	The K2 portal where you may log product support requests
K2 Center of Excellence	A collection of resources to help organizations implement BPM using K2

Questions, Comments or Feedback about this training course?

Use of this content is subject to the Terms of Use. Please E-mail learning@k2.com with your comments or feedback. We appreciate any feedback that helps us to improve the quality of our learning material.

100.ABZ: Introduction to K2



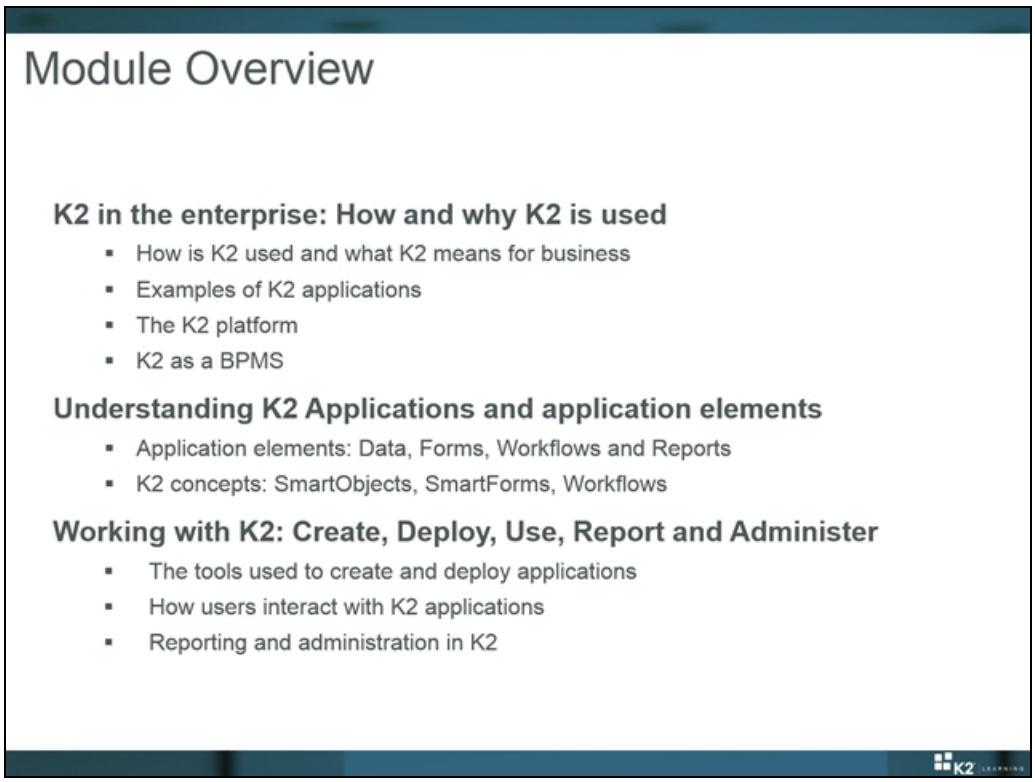
Video

The *100.ABZ: Introduction to K2* training module is a high-level introduction to K2 and covers the following:

- Explains how and why K2 is used by enterprises
- Overview of the K2 platform components and products
- Explains the core elements of K2 applications (Data, Forms, Workflows, Reports)
- Describes the tools used to Build, Deploy, Use, Report and Administer applications in K2

This is a very high-level introduction and overview of K2, and there are no practical hands-on exercises or demos in this module. However, we will be describing some core components and uses of the K2 platform that help to set the scene for the next series of learning modules in this course.

Module Overview



Video

This learning module consists of three parts:

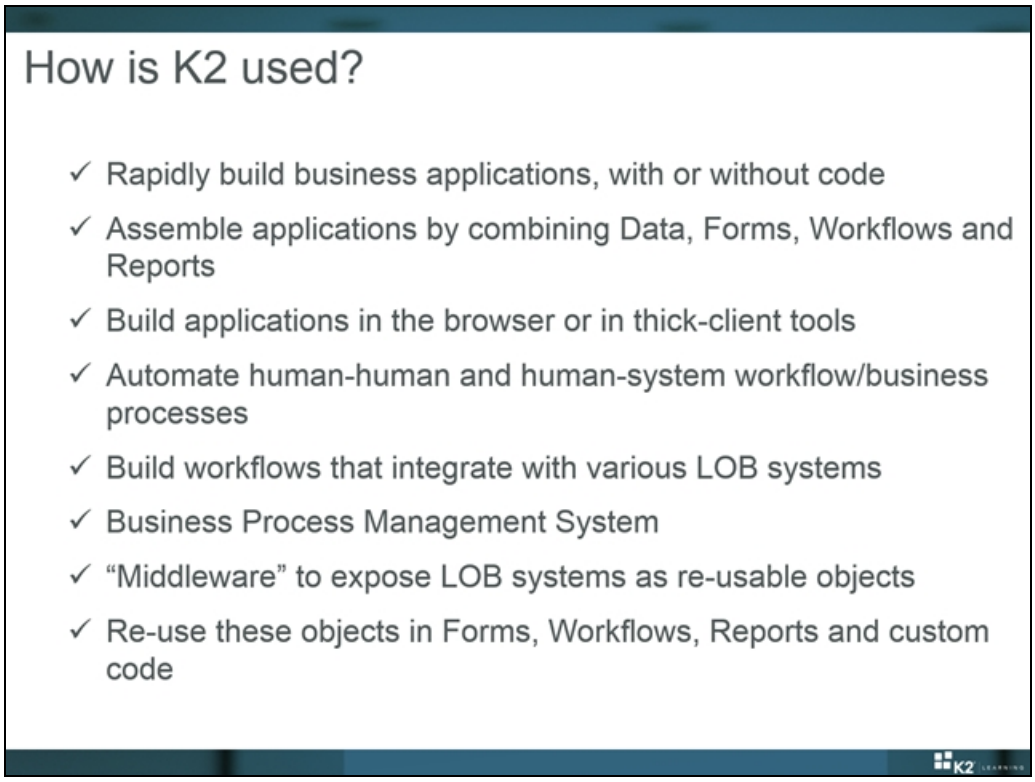
- Part 1 deals with how K2 is used in organizations and provides an overview of the platform as a whole
- Part 2 deals with K2 applications in terms of Data, Forms, Workflows and Reports
- Part 3 describes the various tools and interfaces used to work with K2, in terms of the Create > Deploy > Use > Report > Administer lifecycle

K2 in the enterprise

Video

Part 1 of this learning module is focused on understanding why and how K2 is used in the enterprise. We will discuss typical applications built with K2, the components of the K2 platform and how they are used, benefits of using K2 and some considerations when you want to implement workflows with K2.

How is K2 used?



Video

K2 is used by organizations to implement applications or projects. K2 applications can be developed without writing any code, in effect creating a platform that allows power users to assemble application with Data, Forms, Workflows and Reports themselves, without having to register a development project with IT. That said, K2 can also be used by development teams that write code to implement larger projects with tools like Visual Studio. The platform supports both styles of applications.

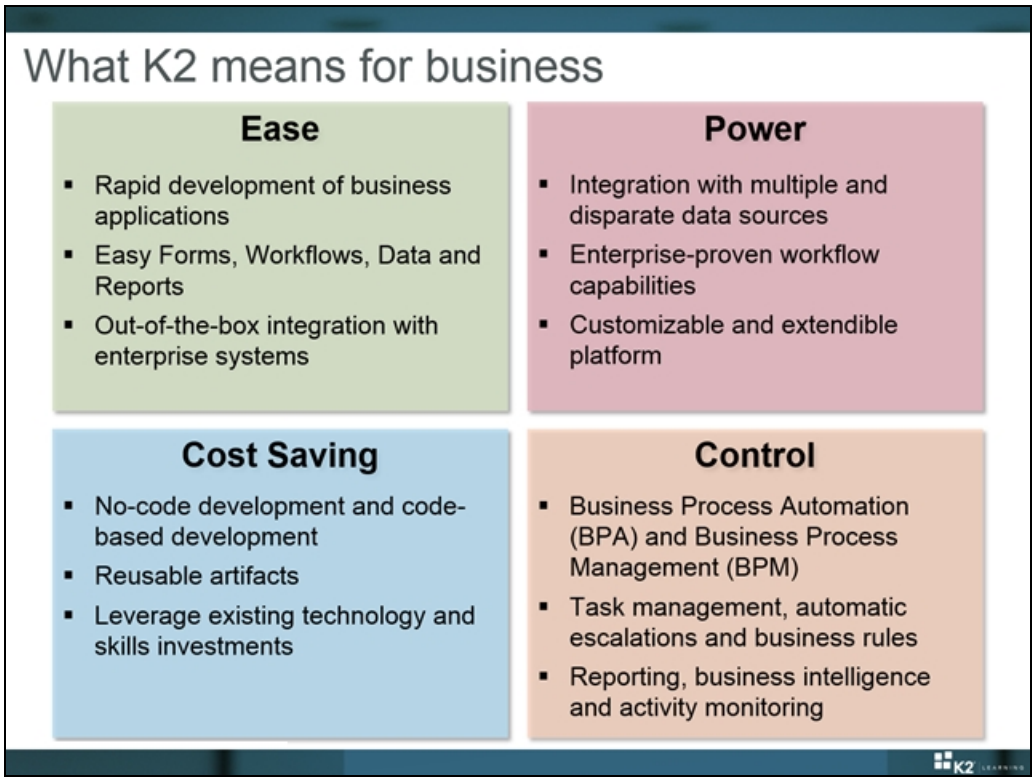
K2 provides tools for both the power user audience (such as the browser-based K2 Designer environment) and tools for developer audiences (such as K2's integration with Visual Studio). You can integrate with many systems using K2's proprietary SmartObjects technology, and build powerful workflows that automate human-to-human and human-to-system business processes.

It is also possible to re-use many artifacts from K2 in other applications. This allows an organization to build up a repository of "business entities" or re-usable workflows, forms and reports which can easily be used by other teams or power users when they assemble their own applications.

Summary

- K2 is a platform used to build business applications
- Applications consist of a combination of Data, Forms, Workflows and Reports (not all the components are required in all applications)
- You can build applications with or without code
- Designers/developers can use browser-based tools (like K2 Designer) or thick-client tools (Like K2 Studio or Visual Studio) to build application elements like Forms, Workflows, Data (SmartObjects) and Reports.

What K2 means for business



Video

K2 meets several business challenges and helps solutions designers and developers create business applications that solve common issues faced by businesses. To understand how K2 meets these challenges, we will categorize them as follows: Ease, Power, Cost Saving and Control.

Ease

Rapid development of process-driven applications

The visual design tools and reusable nature of K2 artifacts empower designers and developers to easily assemble business applications that span Forms, Data, Workflows, Reports and People. These design environments even allow non-developers to create business applications without developer assistance.

Easy Forms, workflow, data and reports

The design tools make it easy for non-developers and developers alike to create Forms, workflows, data entities and reports, reducing the complexity of business applications and time-to-deployment.

Out-of-the-Box integration with enterprise systems

K2 has powerful integration capabilities with standard enterprise systems like Active Directory, Microsoft SharePoint, Microsoft SQL server and SSRS, Oracle databases, Microsoft Dynamics CRM, SAP and more. These standard integration features make it very easy to develop business applications that seamlessly and transparently span across different technologies without having to write any custom connectors or adapters.

Power

Integration with multiple and disparate data sources

K2 SmartObjects make is possible to integrate seamlessly with multiple and disparate data sources. Designers can create logical business objects that combine data from different systems and then use these entities in their business applications.

Enterprise-proven workflow capabilities

The K2 platform features a powerful and proven workflow platform. This platform has proven itself capable of handling everything from simple, single-item departmental level workflows to mission-critical business processes that span multiple business units and systems.

Customizable and extendible platform

The K2 platform is built with customization and extensibility in mind. This makes it possible for developers to create plug-ins that expand the standard out-of-the-box functionality provided by K2. Developers can also write custom code that leverages the built-in power of the K2 platform in their own bespoke applications.

Cost Saving

No-code development and code-based development

The K2 platform makes it possible for non-developers to create business applications without writing any code. This has significant cost savings since it reduces the time taken and resources required to deploy business applications. That said, development teams can use their existing knowledge of the Microsoft .NET framework to integrate with K2 and extend the capabilities offered by K2.

Reusable artifacts

Many K2 artifacts can be re-used in multiple applications which can also help to reduce the cost associated with creating business applications. K2 SmartObjects can be used as a repository of functional business entities that can be re-used in different applications, while K2 workflows can be designed and implemented as re-usable processes.

Leverage existing technology investments

K2 runs on top of Microsoft enterprise technologies and the platform itself is built on the Microsoft .NET framework. This means that organizations can leverage their existing investments in technologies and people skills when using the K2 platform to create business applications.

One Environment for multiple applications

The same K2 environment and infrastructure can be used to implement different business applications of varying levels of complexity; it is not necessary to implement separate K2 environments for each application you wish to implement on the platform.

Control

Business Process Automation (BPA) and Business Process Management (BPM)

The workflow engine in the K2 platform enables Business Process Automation to automate and control the execution of workflows. The K2 platform itself also offers features like rich reporting and administration capabilities to ensure that the core principles that underlie Business Process Management (BPM) can be applied as well.

Task management, automatic Escalations and Business Rules

The built-in Task Management, automatic Escalation handling and support for simple or complex Business Rules (whether defined internally or externally) allows the organization to control their processes and ensure that work is completed correctly, on time, effectively and by the right people.

Many business processes have complex task routing, assignment and management requirements, and the K2 platform is specifically developed to cater for these requirements. Everything from automatic time-based redirection, Out-of-Office task rerouting, dynamic participant resolution to voting scenarios and complex outcomes are supported in the K2 platform. Support for dynamic business rules is typical in many processes. K2 processes can query business rules stores for the latest rule configuration, and route or execute processes accordingly.

Reporting, Business Intelligence and Activity Monitoring

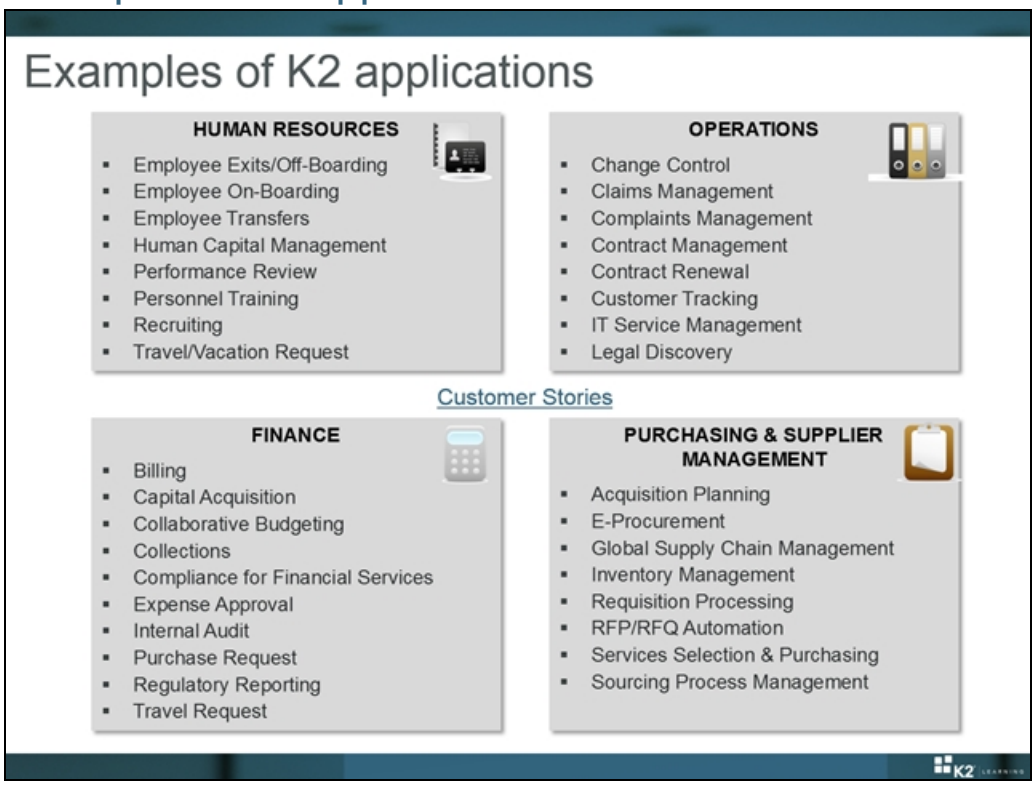
K2 offers rich reporting, Business Intelligence and Activity Monitoring which ensures the visibility of business processes. The reporting platform is also extensible and an organization may create their own custom reports that leverage the standard reporting data (which is automatically captured by K2) to build relevant reports, which may also include data from other systems.

Process reporting information is stored indefinitely (and can be archived, if required), and process security ensure that only authorized users can report on the processes they are allowed to view.

Summary

- There are many benefits to using K2 to implement applications and business process-centric solutions in an organization
- K2 provides ease, power, cost saving and control to the organization

Examples of K2 applications



Video

There are several types of applications that are typically implemented in K2, just some of which are listed in the slide for this topic. Some of these examples may be relevant to your organization, and you can probably think of many more examples specific to your organization or industry. These applications may or may not have a business process at their core: you can certainly use K2 to build applications that have no workflow component.

Business processes could be driven from core business activities (such as order entry and production) and supporting business activities (such as administrative processes). They may also be driven by external forces, such as industry standards or industry requirements (for example change management/change control), or legislation (for example, laws governing financial records management processes).

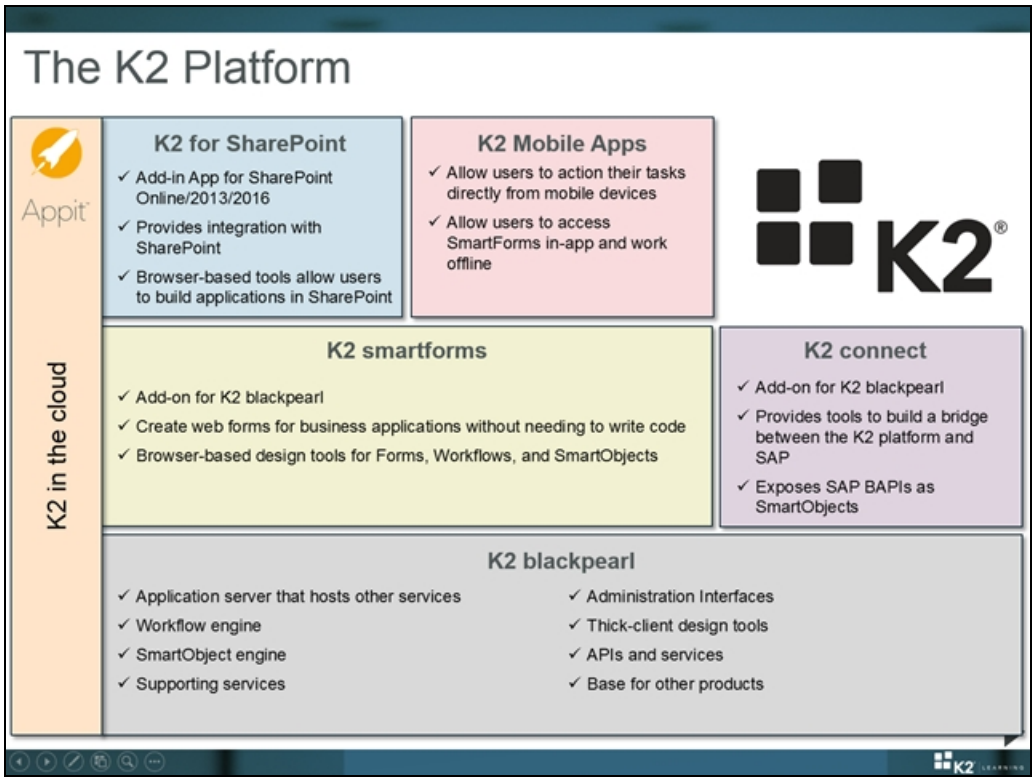
In short, there are many scenarios where a platform such as K2 can help to address business process requirements, and K2 has already proven itself worldwide in all the examples mentioned above plus many, many more.

To learn more about some real-world implementations of the K2 platform, visit our K2 Customer Stories web page at http://www.k2.com/customers

Summary

- K2 is a platform used to build business applications. (K2 does not currently come pre-installed with applications as part of the product, because business process-based applications are different between organizations and industries.)
- These applications may or may not have a workflow component, and may or may not integrate with other systems
- This is not an exhaustive list by any means and there are many other examples of applications built with K2
- K2 is a platform that allows you to build applications like the ones listed here.

The K2 Platform



Video

The K2 platform consists of several products that work together to allow organizations to build powerful applications. Not all of the products are required, although K2 blackpearl is the basis that other components like K2 connect and K2 smartforms build on.

Let's briefly look at the main features of the various products in the K2 platform.

K2 blackpearl

K2 blackpearl is an extensive, feature-rich and powerful platform for building process-driven applications across enterprises. With features like extensibility, powerful integration capability and a scalable architecture, K2 blackpearl is appropriate for organizations ranging from small to global enterprises. K2 blackpearl is effectively an application server that runs several services, including the K2 workflow engine, SmartObjects engine and various supporting services like authentication and logging. You not even need to other components to be able to leverage K2's powerful enterprise features: developers can use the available APIs and Services along with the thick-client design tools to build powerful applications featuring workflows and SmartObjects.

K2 blackpearl's main features are:

Create process-driven applications, fast

The visual design tools provided with K2 blackpearl allow process designers and developers to rapidly assemble process-driven applications that integrate with many business systems, all without requiring code.

Enterprise-capable workflow engine

K2 blackpearl is an enterprise-level platform and supports distributed installation, scaling and load-balancing. K2 can also leverage the capabilities of the Microsoft Windows Server platform and SQL Server, both of which can be configured to support massive enterprise-level implementations.

Powerful task management

Enterprise-capable workflow engine. K2 blackpearl has built-in support for simple and complex workflow task management, such as support for routing work to users, groups and roles, redirection and delegation of tasks, support for mobile task lists, Out-of-Office capabilities and even support for more complex user task assignment, such as voting scenarios or dynamic evaluation of task participants. The escalation capability provided with K2 blackpearl helps to ensure that tasks are completed on-time and within SLA requirements.

Reporting, Business Intelligence and Business Activity Monitoring

Reporting and BI are very important in any enterprise-level workflow platform. K2 blackpearl automatically stores workflow history and auditing information, and exposes this information through standard reporting capabilities and also supports custom reporting to meet BI requirements. These reports are exposed through standard web-based interfaces, or can even be integrated into other applications for a seamless user experience. In addition, the K2 SmartObjects component allows for custom ADO.NET-based reports to combine reporting data from workflow and external systems into one report. K2's workflow engine offers real-time visibility for running processes, allowing organizations to perform real-time Business Activity Monitoring.

Extendible and customizable

K2 blackpearl's pluggable architecture allows organizations to extend the platform to meet business or technology requirements (for example, adding support for non-standard authentication mechanisms, or creating custom wizards to allow process designers to integrate with bespoke systems without writing code). The customization capabilities allow for configuration-based tweaking of the product components to meet requirements, for example changing the logging output or using K2 APIs to write custom applications that integrate with K2 blackpearl. The provided APIs expose K2 blackpearl so that developers have free reign to use code (if required) in workflows, to build user interfaces, implement reporting and BI requirements, build system management and administration utilities and more.

Extensive integration capabilities

K2 blackpearl has extensive integration capabilities. K2 provides several out-of-the-box wizards and services that allow the platform to integrate with various common Microsoft enterprise technologies such as Microsoft SharePoint, Active Directory, Exchange Server, Dynamics CRM, SQL Server and Reporting Services. K2 blackpearl also offers standard support for Salesforce, web services and more, and the K2 SmartObjects technology can enable K2 blackpearl to integrate with almost any other technology. K2 blackpearl can also expose itself as a resource to other enterprise systems through standard interfaces such as SOA-based interfaces, so that the platform can be integrated with other solutions and systems as well. This makes it easy for developers to consume K2 blackpearl in their applications and leverage the capabilities offered by K2 blackpearl.

K2 smartforms

K2 smartforms is an add-on for the K2 blackpearl product and is used to create forms (user interfaces) for business applications without writing any code. Because SmartForms are built on the K2 platform, they integrate very easily with K2 workflows and/or K2 SmartObjects and designers/developers can create very powerful business applications without writing any code.

Create web-based business forms without writing code

The main feature of SmartForms is the ability for both developers and non-developers to create web-based user interfaces for business applications without writing any code. The visual design environment lets designers create complex form behavior through wizard-based configuration and configurable "programming" statements.

Simple, web-based drag and drop form design environment

SmartForms are designed with a web-based visual design environment and do not require any software installation on the designer/developer's workstation. This design environment uses drag-and-drop and wizards during the design process, making it easy to assemble user interfaces without the need for code.

Integrate forms with K2 Workflows and K2 SmartObjects

Because SmartForms are built on the K2 platform, they can integrate easily with K2 Workflows and K2 SmartObjects. This means that designers/developers can easily create user interfaces that leverage the powerful integration capabilities of SmartObjects to interact with almost any back-end data provider, and optionally integrate the user interfaces with the proven K2 Workflow engine for business processes.

Expose and interact with almost any back-end system using a SmartForm

Because SmartForms use SmartObjects for data access, any system exposed as a K2 SmartObject can be used on a SmartForm. K2 provides a range of standard adapters for common enterprise systems such as Microsoft SharePoint, Active Directory, Dynamics CRM, SQL Server, Salesforce, Oracle databases and more, and the platform can be extended with custom adapters for non-supported back-end systems. If it is not possible to expose a back-end system as a SmartObject, developers can also create SmartForm custom controls to integrate directly with a back-end system, skipping the SmartObjects layer.

K2 for SharePoint

K2 for SharePoint is an add-in App for SharePoint, and is a provider-hosted App that provides integration and application design capabilities for SharePoint 2013 and SharePoint online. K2 for SharePoint is part of K2 blackpearl, and it allows you to:

Build SharePoint-centric applications using the browser

Build SharePoint workflow-centric applications directly in the browser.

Build powerful business applications in SharePoint that combine Data, Forms, Workflows and Reports

Build complex applications within SharePoint that can combine external data with customized forms and powerful workflows.

Build applications in K2 blackpearl (e.g. K2 Studio) that integrate with SharePoint 2013

Build K2 or external applications that leverage K2's integration with SharePoint.

K2 mobile apps

The K2 mobile applications are mobile platform applications for operating systems like iOS and Windows devices, and it allows end users to open and action their K2 tasks directly from the app. In addition. users can access SmartForms directly from the app and even work offline with the forms, as long as the forms have been configured to allow for this. Of course, users can still access web-based components from browsers on any device, but the mobile apps just add additional features like batching task actions, working offline and working with K2 in-app as opposed to through a browser.

K2 connect

K2 connect is an add-on for the K2 blackpearl product and is used to expose the widely-adopted SAP enterprise application platform as a data source using K2 SmartObjects. This allows an organization's SAP environments and entities (BAPIs) to be integrated into applications without writing code. Once the BAPIs have been exposed as K2 SmartObjects, solution designers and developers are able to consume these SmartObjects in their workflows, SmartForms, reports, custom code and other applications that consume K2 artifacts.

Expose SAP systems as K2 SmartObjects

K2 connect is a middleware layer which allow organizations to expose SAP entities and information as K2 SmartObjects. K2 connect can interact with most standard and custom SAP BAPIs.

Visual Design tools

K2 connect exposes Visual Design tools to help developers create business entities that reference information in SAP. While developers should be familiar with the SAP BAPIs used in the SAP environment, no coding is required to design business objects that expose SAP entities and BAPIs.

Leverage SAP information across applications

Using K2 connect, designers can integrate SAP information into workflows, user interfaces, reports and more, all without writing code. K2 connect can even be used to integrate SAP into other systems via the K2 SmartObjects technology.

K2 Appit

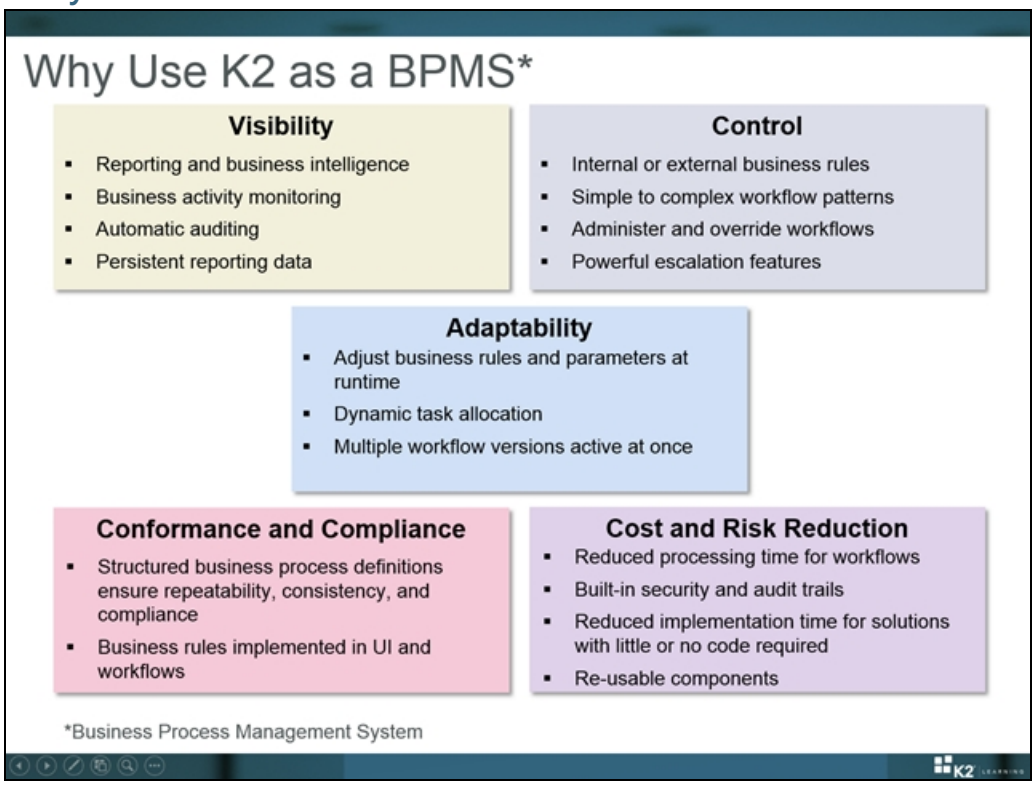
K2 Appit for SharePoint is a cloud-hosted version of the K2 platform that allows you to provision a K2 environment in the cloud that connects to your SharePoint (online or on-premises) environment.

Summary

- K2 blackpearl is the "base" that provides all of the workflow and integration capabilities. It can be used as a stand-alone by organizations who choose to use code to build applications on top of a powerful integration and workflow platform.
- K2 smartforms is an add-on component that allows you to build Forms (and SmartObjects and Workflows) using web-based design tools.
- K2 for SharePoint is an add-in ("app") for SharePoint that allows K2 to interact with SharePoint 2013.
- The K2 mobile apps allow end users to work with K2 tasks and K2 smartforms from their devices
- K2 connect is an add-on that allows you to expose SAP BAPIs as SmartObjects.

- K2 Appit for SharePoint is a cloud-hosted platform that allows you to provision a K2 environment in the cloud that connects to your SharePoint (online or on-premises) environment.

Why use K2 as a BPMS



Video

You may be wondering why an organization would want to use a platform like K2 to implement their BPM and BPA requirements. When implemented correctly, BPM and BPA can provide valuable benefits to an organization in terms of visibility, control, adaptability, conformance and compliance and the reduction of cost and risk.

Let's look at each of these to understand how K2 helps an organization meet these requirements.

Visibility

Visibility of processes, includes monitoring active processes (Business Activity Monitoring) and reporting on historical process data (Business Intelligence). Visibility is often very difficult to achieve in a manual, paper-based process, but once a process is automated, it should be easy to view and report on the status of a workflow at any time. K2 offers powerful Reporting, Business Intelligence and Activity Monitoring features. K2 also includes automatic auditing of events in workflows, and reporting data is stored persistently. In fact, K2 will store reporting data for processes permanently, but it is possible to archive reporting data if necessary.

Control

Businesses need to be in control of their processes. This control includes the ability for authorized users to adjust the security around processes, manage task allocation and manage active processes, but also for the management of business rules that affect the process. In short, a BPM/BPA solution should not make it more difficult or impossible for a business to manage their processes. Quite the opposite: a well-implemented automated process should make it easy for a business to control the process, but not at the expense of conformance and compliance.

K2's support for internal or external business rules make it possible for a business to define business rules where it makes sense. The workflow engine in K2 supports everything from simple, linear workflows to complex workflows with multiple escalations, approvals, dynamic routing, sub-processes and more.

If necessary, authorized users can administer and override workflows at runtime to handle exception conditions, and the platform offers powerful escalation capabilities to ensure that a process instance doesn't "fall through the cracks".

Adaptability

A well-implemented BPA solution should allow the process to adapt to changing business requirements and evolve with the business when necessary. The implementation and architecture of a solution should make it possible for a

business to change the behavior of a process (where relevant) so that the business can respond to changing conditions rapidly, but not lose control or risk compliance failure in a process. Here is an example: a new customer on-boarding process may require a series of approvals when discounts are offered to new customers. Perhaps a competing business is offering new customers a discount without requiring a long approval cycle. Your organization should be able to adapt their internal process to retain an advantage over the competitor, but not lose control over any substantial discount approvals that may be required. A simple way to implement this approach in a business process is to specify the minimum discount percentage that requires approval in an external business rules table. If the organization needs to adjust that minimum percentage in response to a changing market, the value can be adjusted without requiring any change to the business process itself. K2's support for looking up business rules in external sources at runtime makes it possible to adjust these rules on-the-fly without affecting the workflow design, making the process very adaptable.

K2 also has powerful dynamic task allocation features to ensure that the right work is delivered to the right people on time. You could leverage the information already defined in your organization's Active Directory environment for routing, use roles and groups defined in external systems or use the Role feature in K2 to maintain role membership. Tasks can even be routed in round-robin or other task allocation algorithms, if needed.

The K2 Workflow engine also makes it possible to execute multiple versions of a workflow simultaneously. You can choose whether to promote, demote or delete workflow versions and even migrate in-flight workflows from one version to another.

Conformance and Compliance

Automating a business process should make it easier for the organization to conform to business directives and strategies and to comply with standards, guidelines or requirements. For example, a BPM solution could help an organization to conform to best practices for product change control, or to comply with legislative requirements for prescribed processes. Take for example an organization that manufactures medications. A well-designed and implemented Product Change Management process is crucial to this business, not only to protect the organization from legal and ethical risks and to ensure that production remains on-line, but also to conform to legislative requirements that may prescribe the process that should be followed when a controlled product is changed. Automating processes using BPA will remove a lot of the uncertainty, ambiguity, opacity and scope for misuse or subversion that are inherent in manual processes.

Structured business process definitions like the ones implemented in K2 ensure that processes are repeatable, consistent and in compliance with business directives. The business rules that govern these processes could be implemented in the underlying workflows or in the user interfaces used to interact with those workflows.

Cost and Risk Reduction

Effective and efficient business processes almost always bring reductions in cost to the business. Several case studies have been published where organizations have seen substantial cost savings by implementing BPM and BPA solutions, especially when moving from inefficient paper-based processes to fully automated processes.

In addition to cost savings, automated processes also help to reduce risk, since automatic processes can remove many human risk factors, and should provide a full audit trail of processes. Conformance to industry standards or legislation (and crucially, proof that the organization did comply or conform to these requirements) can also reduce an organization's liability. K2 features built-in authorization and auditing to help organizations reduce these risks.

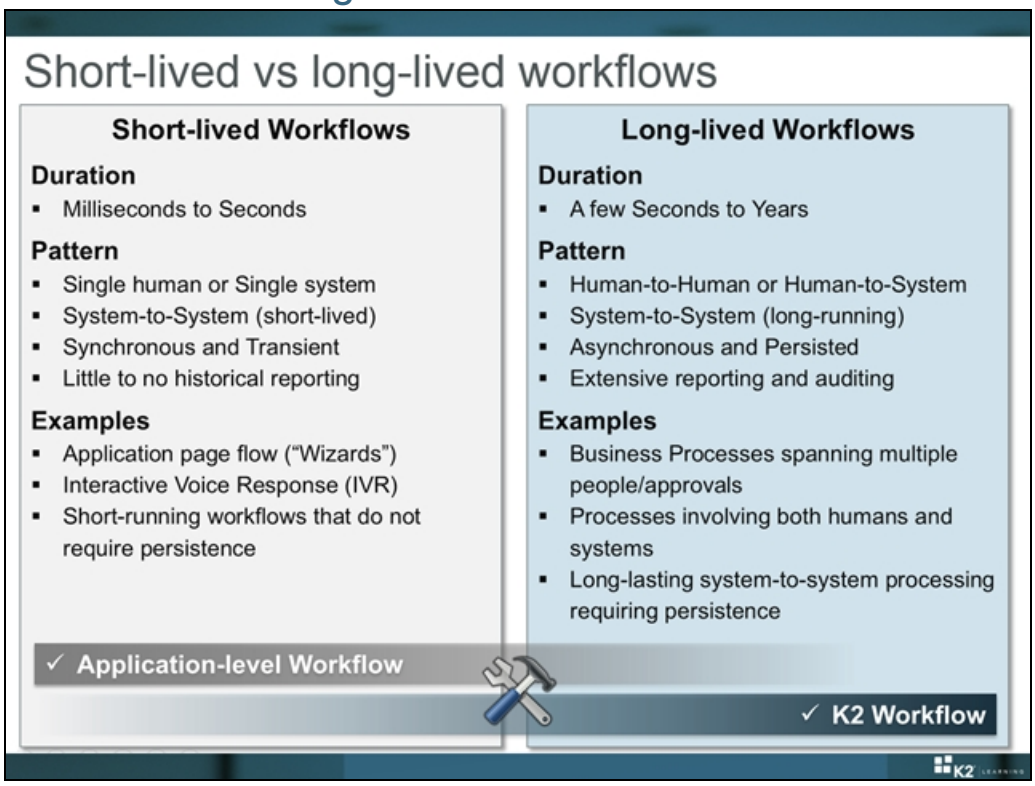
Apart from the inherent benefits brought by an automated process, the development tools available in K2 also help to reduce costs and risks since it is possible to create business applications without writing any code. K2 artifacts and components are also re-usable across different projects, making it faster, cheaper and easier to implement applications that integrate seamlessly with existing technologies and enterprise systems. Even so, it possible for a developer to write code against the K2 platform and use the powerful features like the workflow engine in their own custom applications, saving delivery time for more advanced solutions.

Summary

- K2 is perhaps best known as a “workflow” engine but it is so much more than just a workflow engine. It is a platform for building business applications.

- That said, many organizations primarily use K2 as a Business Process Management System (BPMS) since it has a very robust and powerful workflow engine and supporting components like integration and built-in reporting, that allow it to solve many Business Process Automation and Business Process Management requirements.
- The main benefit of using K2 as a BPMS is the power of the platform and the control it gives organizations over their processes.
- Common Terminology:
  - **Workflow/Business Process**: Sequence of connected steps, procedures or activities that may span across different people, roles, departments, organizations or systems. Often visualized with a flowchart. In K2 terms, a “workflow” and a “business process” are essentially the same thing.
  - **Business Process Automation (BPA)**: Using software or a platform to implement automated workflows/business processes.
  - **Business Process Management (BPM)**: Optimization of business processes using management strategies, workflow formalization, BPA, rules, engines and organizational structures. Business Process Management is concerned with the lifecycle of the Process Definition.
  - **Business Process Management System (BPMS)**: Technology or platform used to implement BPM and BPA.
  - **Business Process Modeling Notation (BPMN)**: A “language” for visually describing processes. K2 does not rely on nor explicitly use BPMN but it is useful to use BPMN when designing workflows. At time of writing there are no standard tools that allow importing BPMN diagrams as K2 workflows or vice-versa.
  - **K2 Business applications**: Bind Forms, Reports, Data, Workflows and people together using the K2 platform and its integration features.

Short-lived vs long-lived workflows



Video

The term "workflow" is actually a little ambiguous, since it could refer to two types of workflow processing: Short-lived workflows and Long-lived workflows. It is important to understand the difference between these styles of workflow processing, since the K2 platform lends itself better to implementing long-lived workflows.

Short-lived workflows

These are probably best explained though examples: consider an Interactive Voice Response (IVR) process. (You may be familiar with this kind of workflow: when you dial a number and an automated system guides you through several options, each step determined by the preceding step. "Press 1 for sales" directs you to a further option to "Press 1 for order enquiry, Press 2 for new orders" and so on.) This is a typical example of a short-lived, synchronous workflow. The workflow (in this case, the responses to the automated prompts) never leaves the person performing the tasks, and the workflow itself is completed quickly. There is no need to transfer the workflow tasks between people and the workflow usually needs to be responsive with very rapid response times, even under high load.

Short-lived processes are sometimes called application workflows, because the work never leaves the system in which it is being performed. They are also considered to be synchronous, because the tasks happen in sequence and in very short intervals. Usually, these types of workflows do not need to be saved or persisted into a database. Another example of a short-lived workflow is an Enterprise Message Bus implementation, where a system is responsible for transferring work from one system to another, perhaps also performing some message manipulation along the way. This is known as a system-to-system workflow.

Note
Short-lived workflows are often referred to as Transient workflows, since there is no need to persist workflow states for intervals longer than a few seconds or milliseconds. Transient workflows are most often implemented in the application level of a solution, and Windows Workflow Foundation is one of the technologies that can be used to implement this style of workflow.

Long-lived workflows

Now consider a customer onboarding workflow. There are several tasks involved in this process, each of which is performed by a different person in the organization. There is usually also some kind of system integration in this type of process (why re-capture the new customer's details into the financial system, if an automatic process can do it for you?).

The workflow itself may last several days (or even months or years), and the workflow must be persisted for its entire lifetime, rather than just for those few seconds when users are interacting with the process. The workflow history and state are stored in a long-term, persisted storage medium like a SQL database.

Long-lived workflows are often called human-to-human or human-to-system workflows, because the tasks required to complete the process usually move between people and/or systems, and whenever a human is involved in the workflow, it implies that there may be an unpredictable wait state in the workflow. In some cases, system-to-system processes could also be persisted, such as long-running processes where processing is delegated to an external party's system, and it is not known how long that system will take to return the task. The asynchronous nature of these workflows implies that, although we are expecting another step in the workflow, we cannot predict when that step will be completed, and that is why the persistent layer is required.

Note
Long-lived workflows are often referred to as Persisted workflows, since the workflow state must be persisted (stored) between those events where users or systems are interacting with the process. Persisted workflows are most often implemented in some kind of workflow engine like K2.

Why is this distinction important?

The K2 platform is fundamentally designed for running long-lived workflows. All of the components and mechanisms in the K2 workflow engine are built to support the type of asynchronous processing model typical of human-to-human and human-to-system workflows. While you certainly CAN implement transient workflows on the K2 platform, it is not always the recommended approach, since the K2 platform adds additional overhead for persistence which is typically not required in transient workflows. That said, the K2 platform has automation features which can be very useful in system-system workflows, such as creating dynamic document content, integrating with systems like Active Directory or SQL server, sending automated e-mails and more.

It is important for architects and developers to understand when to use which technologies to implement workflow requirements. Software or platforms like Windows Workflow Foundation are specifically written for transient workflow processing where rapid processing time is essential, while the K2 platform is targeted to persistent workflow processing where the state of the workflow must be saved for indeterminate amounts of time. If you are in doubt, make use of K2's support or consulting services to help you make the right choices for your particular requirements.

Tip
The K2 engine can be optimized through various mechanisms to achieve fairly high volumes and a high level of scalability. Application architecture can also affect solution performance.
K2 has published whitepapers detailing the performance and scalability of the platform, and you can find those whitepapers in the K2 Knowledge Center. If you have very rapid response requirements, or are planning to implement high-volume processing on the K2 platform, we encourage you to make use of K2's consulting services to ensure that your solution is architected and configured as optimally as possible.

Questions and considerations that can help to identify the right implementation platform

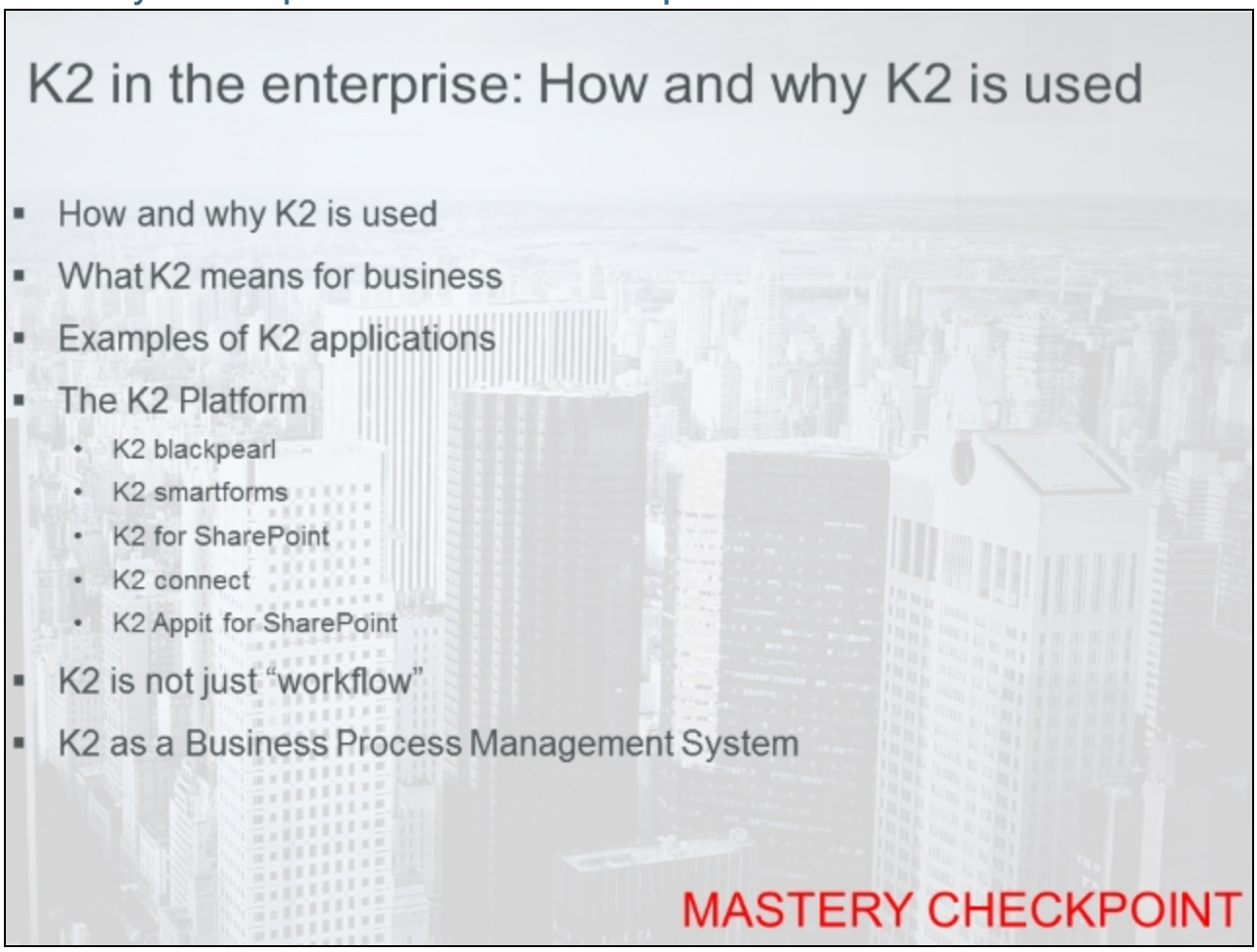
Question	Implementation consideration
Is this a short-lived or a long-lived workflow?	K2's persistence layer makes it more suitable for long-lived workflows. You may use K2 for short-lived workflows if the volume is not too high and the response time requirement is around a half-second or longer.
What is the expected load on the system?	Very high load environments (for example: thousands of activities are expected per second or per minute) may require a dedicated workflow implementation, or may require a more complex K2 architecture to ensure adequate performance.
What are the response time requirements for the system?	Very low response time requirements (for example: response times of less than a quarter of a second) would typically require a dedicated, application-level workflow implementation.
How long do the processes/workflows	Workflows that last a minute or more are likely candidates for a persisted workflow engine like K2.

last?	
Do you need to report on processes that have completed?	K2 automatically stores workflow history and audit information, so consider leveraging this built-in functionality in your solutions: you are effectively getting the workflow history, audit trail and reports for no additional development effort.
Is there a requirement to deliver tasks to users and have a system maintain the task allocation?	Whenever multiple users are involved in a process there is usually a requirement to have some kind of task list or work list where users will access their work items. In addition, when users are involved with a process there is often a requirement to support exception cases (such as Out-of-Office rules or manual overrides) and business rule considerations (such as different levels of approval). Workflow engines like K2 have built-in support for these requirements, so leveraging these capabilities in your solutions will simplify and speed up development.
Should business users be able to design their own processes?	K2 makes it easy for business users to build their own processes using visual design tools. Application-level workflows are often built by programmers in some language like Windows Workflow Foundation, which is not very user-friendly for business users.
Can you leverage the K2 automation and integration components to simplify development for a low-volume, short-lived workflow?	If the expected volumes are not massive, you can certainly make use of the built-in features of the K2 platform to automate repetitive tasks such as dynamic document generation or system integration to create a short-lived workflow. This can save a lot of time when it comes to building new solutions.

Summary

- Often customers want to use K2 to automate non-persisting workflows like IVR or screen flow. This is not the ideal use of the K2 workflow engine.
- While you can implement short-lived workflows with K2, it is not the intended or optimal use of the platform because of the additional overhead required for persisting workflows. K2 workflows are not held “in-memory", which mean they are persisted to a database. This is what causes the additional overhead.
- K2 workflows are better for human-human or human-system workflows, or long-lasting workflows between systems..
- K2 workflows typically last from seconds to years.

Mastery Checkpoint: K2 in the enterprise



Video

This is a checkpoint for the information covered in Part 1 of this module: how and why K2 is used. If you are attending a training session led by a K2 instructor, this is your opportunity to ask questions around K2 application components, SmartObjects, SmartForms and workflows.

These are the main concepts you should understand:

- What K2 means for business
- How and why K2 is used and examples of typical K2 applications
- The K2 Platform/product components at a high level (K2 blackpearl, K2 smartforms, K2 for SharePoint, K2 connect, K2 Appit for SharePoint)
- That K2 is not just "workflow"
- K2 can be used as a Business Process Management System

Knowledge-check questions

To check your own knowledge and comprehension, consider the questions and answers below. Some are specific to K2, others may require you to think how K2 applies to your environment.

Q: Did you recognize any of the sample applications (e.g. employee on-boarding, expense approvals, leave applications) as solutions that might be useful in your own organization?
A: (discussion question, no right/wrong answer)

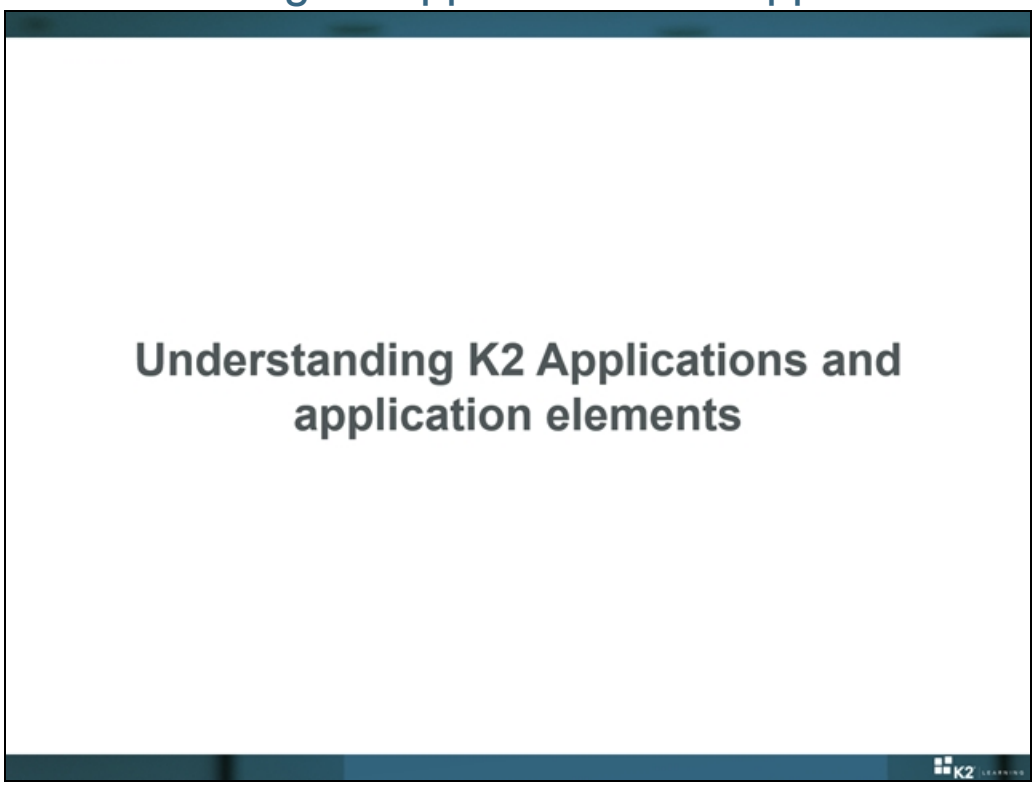
Q: Does your organization have specific goals or projects for K2?
A: (discussion question, no right/wrong answer)

Q: Do you understand the difference between short-lived and long-lived workflows? Can you give an example of each? Do you understand why K2 is better suited for long-lived workflows?
Reveal answer**A:** Short lived: lasts less than seconds, usually application-level workflow (e.g. screen flow or IVR).

Long-lived lasts more than seconds, usually something that binds people and systems together (e.g. approval workflows). K2 is better at long-lived workflows because the workflow state is persisted to a database, which adds processing overhead that could result in low performance for a very time-sensitive applications like IVR

Q: What are K2's strengths as a BPMS? What does it allow organizations to achieve?
Reveal answer
A: Visibility, Control, Adaptability, Conformance and Compliance, Cost and Risk Reduction. With K2, organizations can automate business processes, establish control over processes and easily adapt processes to changing circumstances.

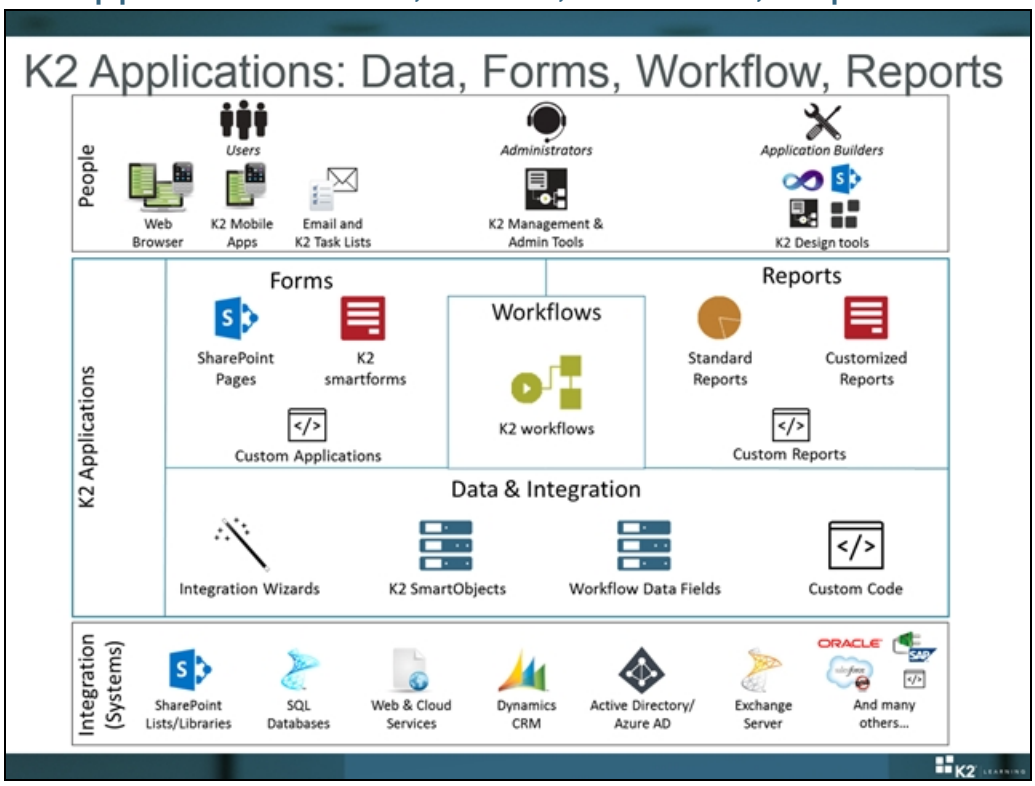
Understanding K2 Applications and application elements



Video

Part 2 of this training module describes the elements that typically make up K2 applications: Data, Forms, Workflows and Reports. We also explain the basic principles behind K2 SmartObjects, K2 smartforms and K2 workflows.

K2 Applications: Data, Forms, Workflow, Reports



Video

The diagram above illustrates the main components of the K2 platform. We have separated these components into **People** (users who work with K2 or use K2 applications), **K2 Applications** elements (the things that make up K2 applications) and **Integration** (the systems that K2 integrates with).

K2 applications consist of four main elements: **Data**, **Forms**, **Workflows** and **Reports**. The application elements fully integrate with each other to provide a robust framework for building applications. However, they also work independently from each other: for instance, you might have a SharePoint List that utilizes SmartObjects and SmartForms, but not Reports or Workflows. K2 also provides tools for designing and customizing the Application elements as well as tools and interfaces to administer the K2 environment, such as managing permissions and managing the active workflows in the environment.

People, Roles and Administering K2 Applications

K2 defines a number of logical roles that make up the human element of your Application. Users are the consumers of your Application's functionality, and they use the Forms that you build to interact with your Application. Usually, users would use your applications with web browsers or mobile applications. Administrators manage workflow instances and permissions, and use K2's Management Pages or other administration tools to perform these functions. Application Builders are the designers and builders of the Forms, Data, Workflows and Reports using the web-based tools such as the K2 Designer.

Forms

Forms are the User Interfaces that users will use to interact with your application. K2 supports a range of technologies for these Forms: K2 smartforms, InfoPath Forms, SharePoint interfaces or custom applications written in .NET or other programming languages. The exact technology used for an application depends on the requirements and the organization's infrastructure, but the main point is that K2 does not bind you to a particular technology for creating forms.

Workflows

Workflows are a logical sequence of steps and events performed by users and/or systems. Workflows provide the "business process" functionality in your application. K2 features a mature and powerful workflow engine which has proven itself over many years in many different organizations.

Consider something like a Leave Request application: in this case, there would be a workflow component which runs the Leave Request-Approval-Processing steps of the application. The workflow assigns tasks to users (the employee's manager, for example, when the Leave Request must be approved) and performs system tasks as well (for example, sending e-mail notifications to the employee to notify them that their Leave Request was approved).

Reports

K2 provides standard reports which expose the metrics of your application workflows, such as the time taken to complete leave requests and audit trails. You can also use SmartForms and SmartObjects to create customized dashboard-style reports, leveraging the available Reporting Controls in the K2 Designer tool. Or, you can build custom reports using third-party reporting tools that are able to consume the available K2 APIs and Services.

Data and Integration

The Data component allows you to integrate your application with providers of data and consumers of data. K2 primarily uses a technology called SmartObjects to interact with back-end systems. You can think of SmartObjects as a "connector" or a "middle layer" that provides the necessary integration to expose the data that resides in some system, to the K2 application elements that need to use that data. The important point is this: SmartObjects are the consistent interface that K2 uses to interact with some provider of Data, regardless of what technology that provider might be. The artifacts in that provider (for example the "Employee" or "Customer" or "Account" entities) are represented as logical business entities that you can easily use within your Applications, without having to know where that data resides.

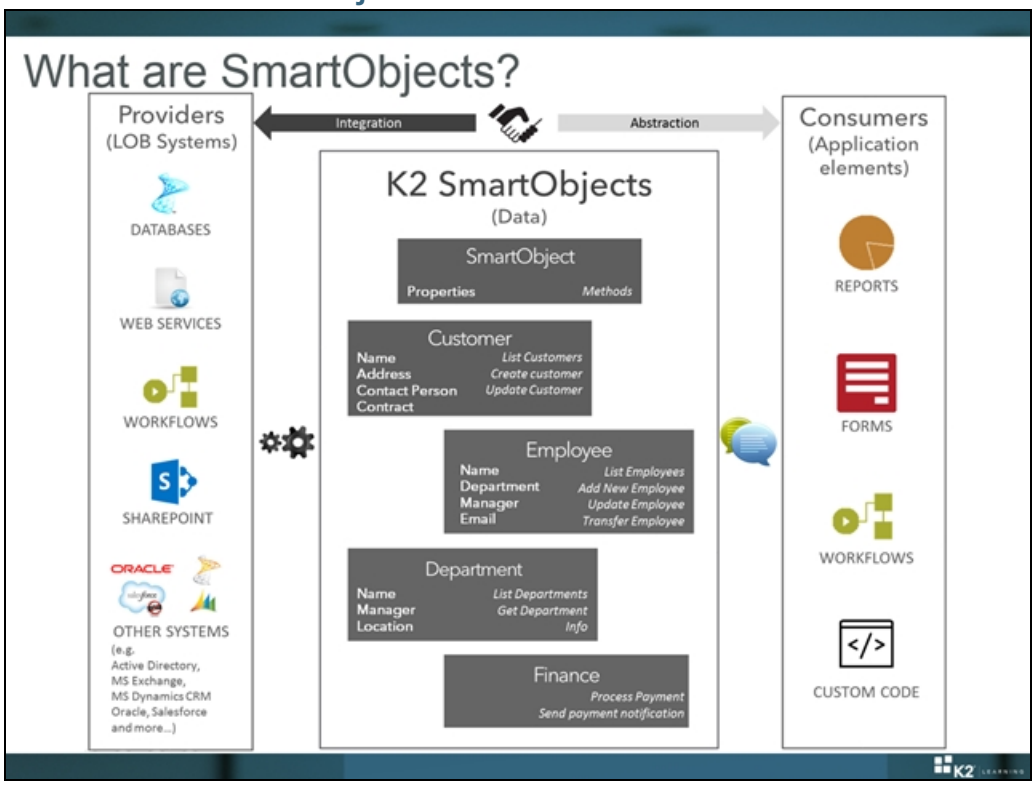
Data could also refer to workflow-level data fields that are defined in workflows, but this is normally limited to storing data that is only applicable to the workflow itself.

K2 provides a set of standard workflow wizards that make it easy to integrate with common enterprise systems like Active Directory, Exchange, SharePoint and many others.

Summary

- K2 applications consist of Forms, Reports, Workflows and Data (Data includes integration with LOB systems).
- The most common options for each components, are SmartForms for the Forms part, SmartObjects for the Data part and standard reports for the Report part.
- SmartObjects are used as a Data Access layer to integrate with other systems.
- Users always interact with Forms or Reports. They would use browsers, K2 applications, email or the K2 task list. Users could also be using custom applications if the organization built their own applications.
- Administrators use the K2 Management Pages to administer K2 environments.
- Application builders use K2 web-based design tools, K2 thick client tools like K2 Studio or tools like K2 for Visual Studio to build applications.

What are SmartObjects?



Video

It is worthwhile to describe what SmartObjects are to understand how they are used. SmartObjects are a very integral part of creating powerful and flexible K2 solutions and are the primary mechanism that K2 uses to integrate with other systems.

Essentially, SmartObjects are a “middle layer” between back-end systems like databases and consuming applications such as user interfaces, reporting engines and workflows. The purpose of the SmartObject component is to allow organizations to create re-usable business objects, which can then be consumed by various technologies. Because SmartObjects are consistent and hide all the integration logic behind the scenes, it is very easy for designers and developers to interact with back-end systems without needing to know anything about those systems.

Consider the scenario illustrated in the slide above. Let’s assume some business data exists in external systems: perhaps employee information exists in a database and a web service provides information about departments in the organization (these back-end systems are shown on the left-hand side of the diagram).

We may want to use this information in some consuming applications (shown on the right-hand side of the diagram). Perhaps an employee’s information must be displayed on a form, or we need a report to list all employees and their departments, or a workflow needs to query an employee so that an approval task can be delivered to the employee’s manager.

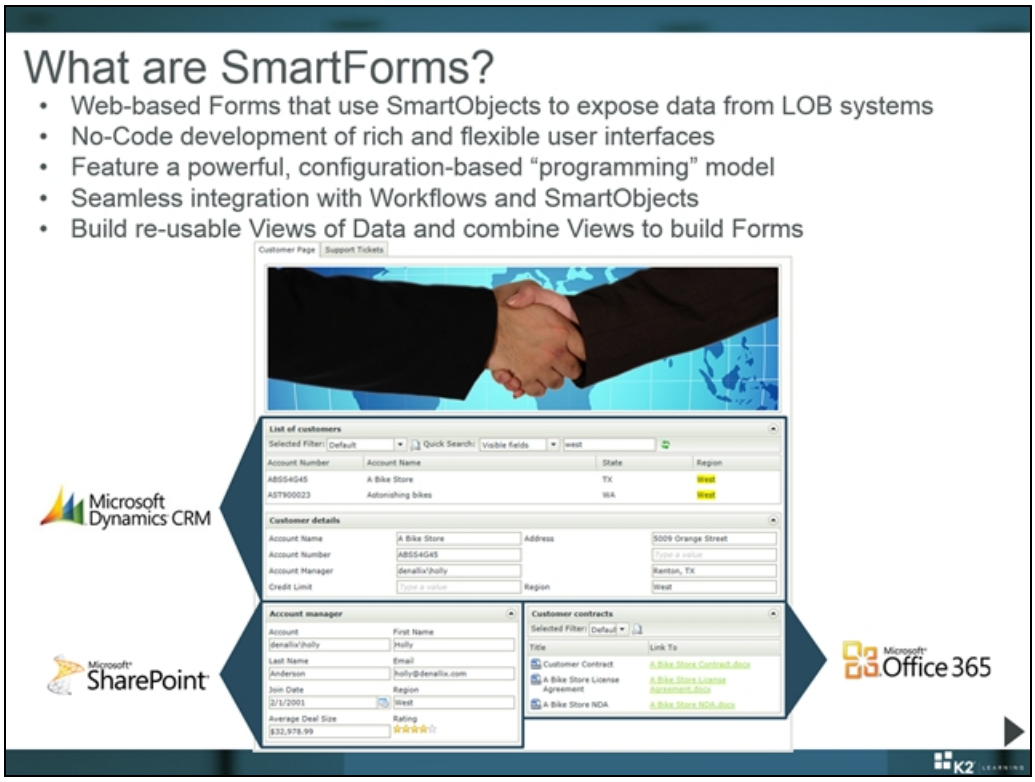
SmartObjects are what make this scenario possible without requiring any code. They are effectively a middle layer between the sources of data on the left and the applications that require the data on the right. The SmartObject component acts like a translator, so that the consuming applications do not need to understand how to query a web service and a database to aggregate information about an employee. Instead, these applications just ask the SmartObject to list employee information, and K2 will retrieve that information from the underlying systems using some technical integration points to access the external system.

Note that the communication is two-way. This means that SmartObjects can be used to both read and write information to the underlying systems. Of course, authentication and authorization mechanisms are in place to ensure that only authorized users are allowed to read or update data.

Summary

- SmartObjects are essentially an abstraction layer that makes it possible to integrate with many different systems as if they were logical business objects.
- In the diagram, “providers” of data are on the left and “consumers” of that data are on the right.
- SmartObjects is a “middle layer” between the providers and the consumers.
- SmartObjects are abstracted representations (“logical business objects”) of the data that resides in LOB systems.
- They make it easy to integrate with various systems because all of the technical integration with various systems is hidden from the consumers, and therefore the users that want to use those objects.
- Typical examples are something like “Employee” where the employee data could come from a SQL database, Active Directory or even a combination of systems.
- SmartObjects allow for both read and write operations.

What are SmartForms?



Video

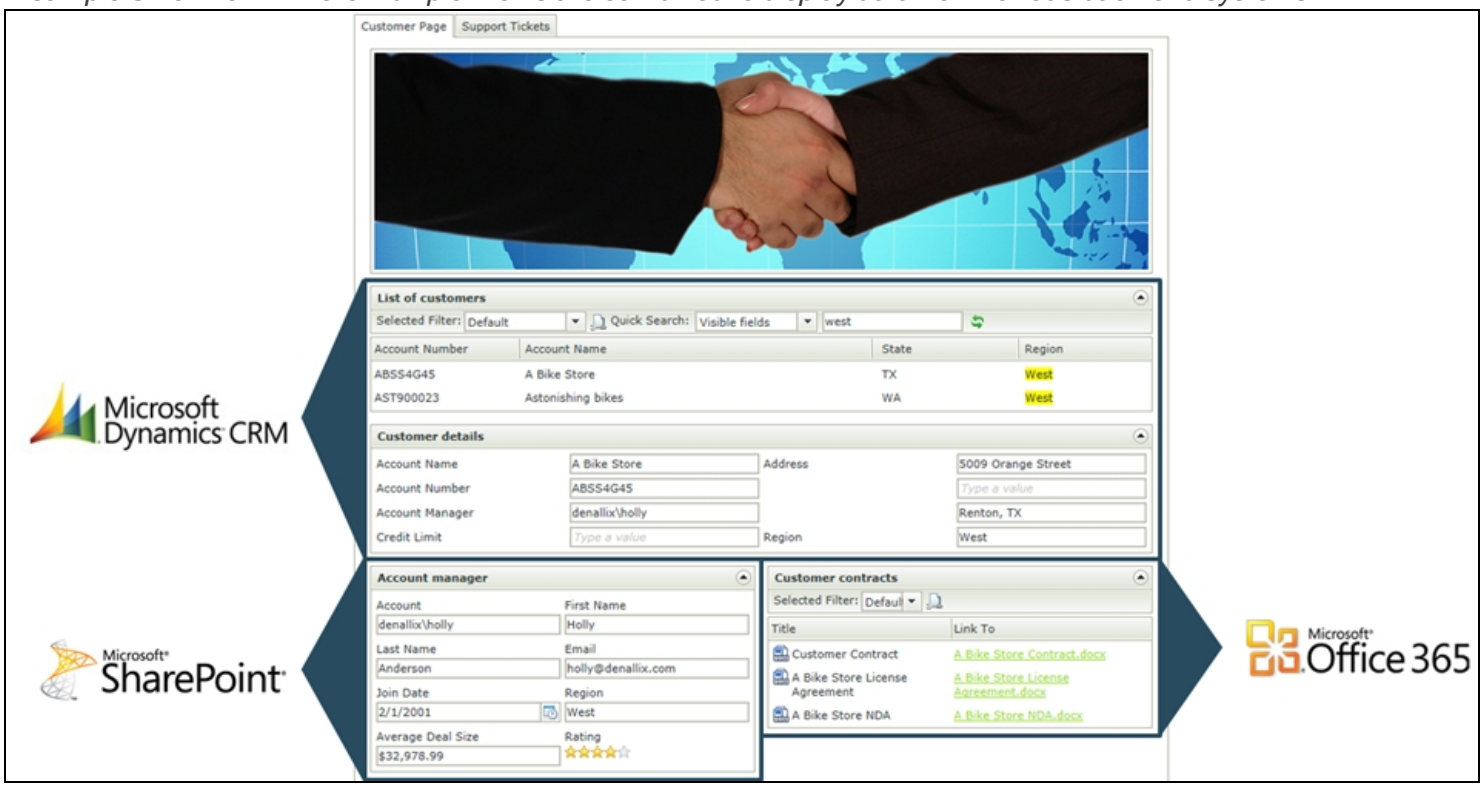
SmartForms are web-based forms that allow users to interact with SmartObjects and workflows. In technical terms, SmartForms are essentially one of the available presentation layers that can be used to build user interfaces for a K2 Application. End users can view and interact with SmartForms through most modern web browsers or through dedicated Apps like the available K2 Mobile Apps. They may also use devices with browser support such as computers, tablets and smartphones to work with these forms. SmartForms are exposed though URLS (which may or may not include query string parameters for additional processing). SmartForms can even be exposed in a SharePoint site using the K2 smartforms Form Viewer Web Part.

K2 smartforms is an add-on for K2 blackpearl, and they integrate closely with SmartObjects and K2 workflows. SmartObjects are used to allow SmartForms to interact with data sources. Any system exposed as a SmartObject can become a data provider for a SmartForm. SmartForms can be as basic as just displaying read-only data on a form, or can be "wired up" to execute SmartObject methods to interact with a data source, including scalar (single-record operations) or List methods (multiple records). It is a very common use case to use SmartForms to capture user input and then save that input to a data store, or to use SmartForms to retrieve lists of information from some back-end system.

SmartForms are built with the K2 Designer tool, a powerful no-code "programming" environment that allows designers to assemble and configure forms with Rules, Events, Controls and Layouts by dragging, dropping, configuring and running through wizards.

Consider the sample Form below. Here, we are using standard K2 service brokers to expose data from Microsoft Dynamics CRM, SharePoint and Office365 as SmartObjects. These SmartObjects have been exposed on re-usable Views, and these have in turn been added to a Form that represents customer information. Ultimately, the user gets a global view of a customer without having to go to different systems. The individual Views can be re-used in other Forms, so if another developer has a requirement to show Account Manager information on a Form, for example, they can just re-use the existing View.

A sample SmartForm where multiple Views are combined to display data from various back-end systems

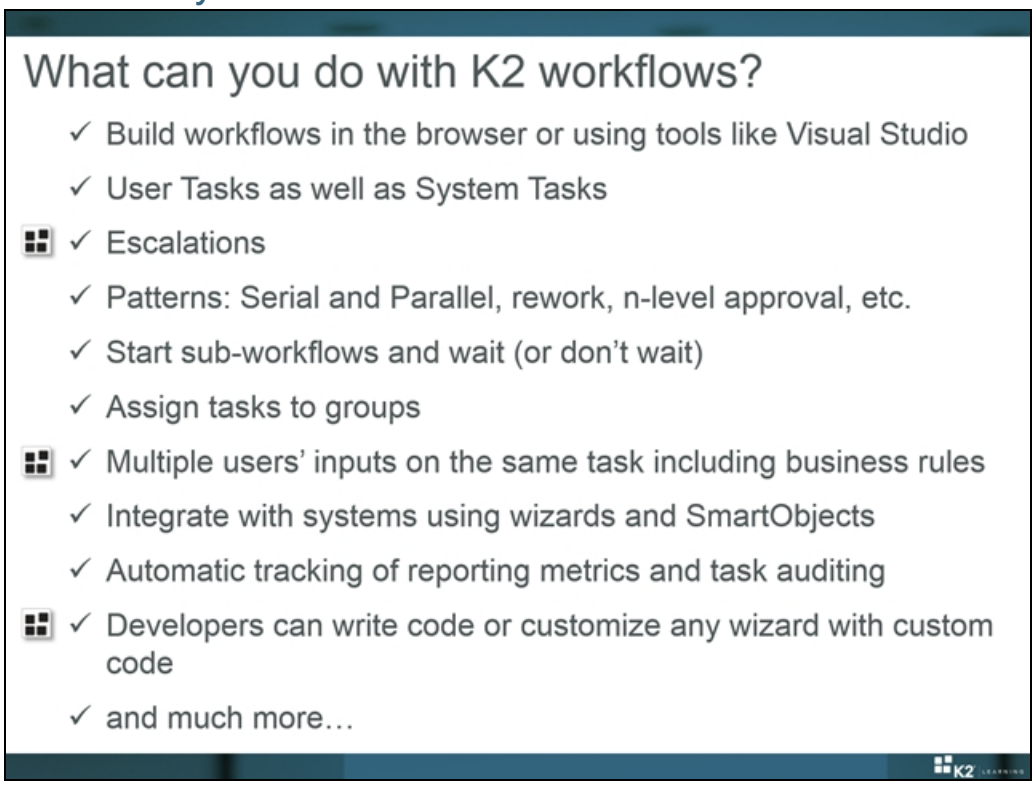


Note
This training course does not cover SmartForms in-depth because SmartForms might not be used by all organizations. Although we will use SmartForms and touch on them lightly during this training course, there is a dedicated training course that covers SmartForms in-depth.

Summary

- SmartForms are web-based user interfaces
- Are developed ("built") using K2 Designer, a no-code, web-based design tool
- SmartForms provide a powerful "programming" model based on configuring rules and actions
- They integrate seamlessly with SmartObjects to expose data from LOB systems
- Designers can build re-usable Views of data and then combine those Views on a Form to arrive at a rich user experience

What can you do with K2 workflows?



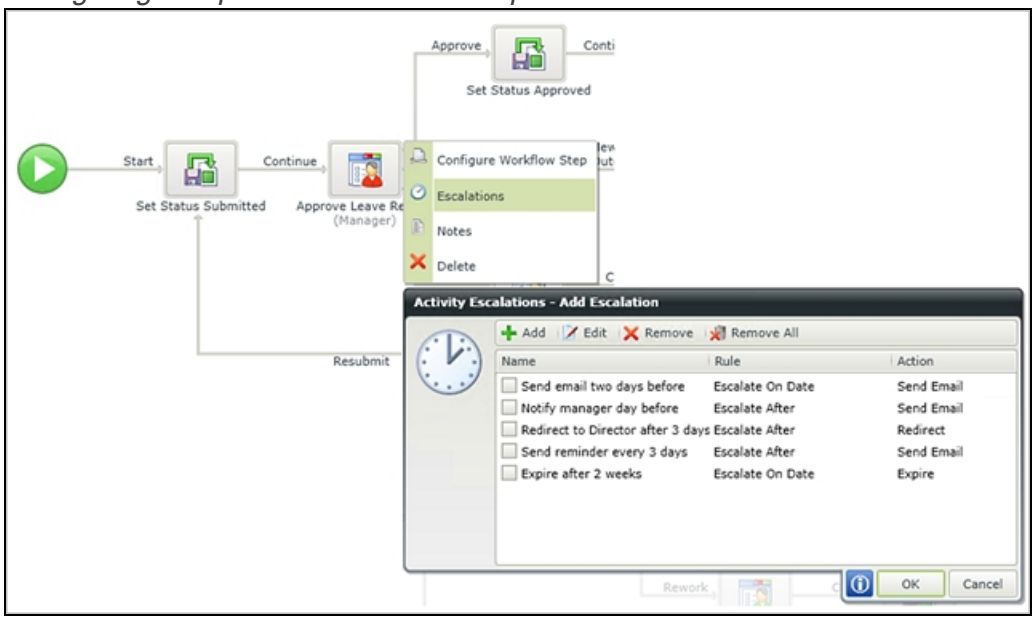
Video

Let's consider some of the common uses of the Workflow component of K2 Applications. (Note that this is just a high-level overview of some of the common uses for workflows, you will learn how to specifically do these things later on in this course.)

Workflows are built using any of several design tools, such as the web-based K2 Designer tool, the proprietary K2 Studio design tool or as an add-on for Microsoft Visual Studio. Workflows support both User Tasks (tasks for humans that wait for input and perhaps a decision like "Approve" or "Reject") and System Tasks (automated tasks like sending emails, updating databases, starting sub-workflows).

Workflows can feature Escalations, for example sending a reminder message when a task is not completed within a specific time.

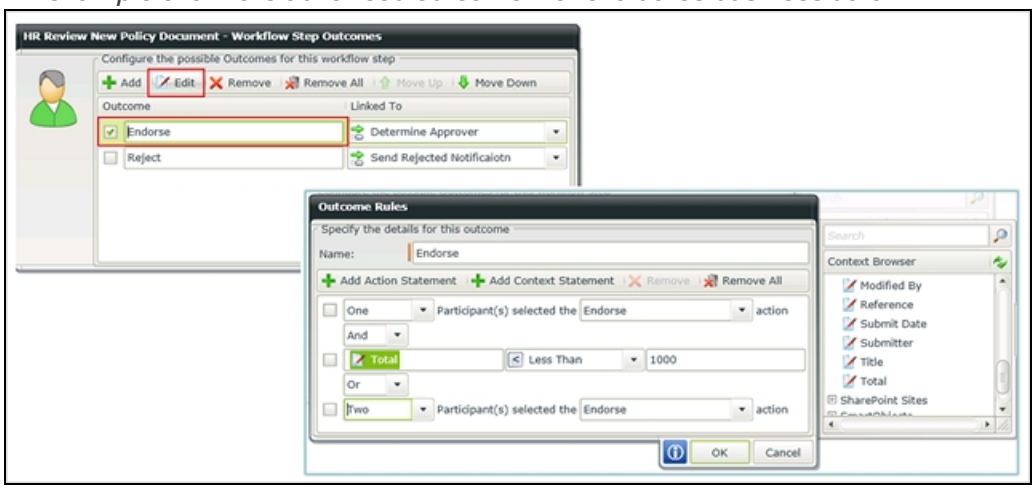
Configuring multiple escalations for a step in a workflow



It is possible to implement many different patterns of workflow in K2, such as Serial and Parallel execution, looping, rework, N-Level approvals and many more. You can also use the Inter-Process Communication (IPC) wizard to start one or more "child" workflows from a parent workflow.

From a task assignment perspective, it is possible to assign tasks to groups of users and then decide how many users in that group need to action the task. You can even extend the outcomes for a task by evaluating business data in conjunction with user inputs, for example if the Total is less than 1000, only one user needs to endorse the request, otherwise at least 2 users need to endorse the request.

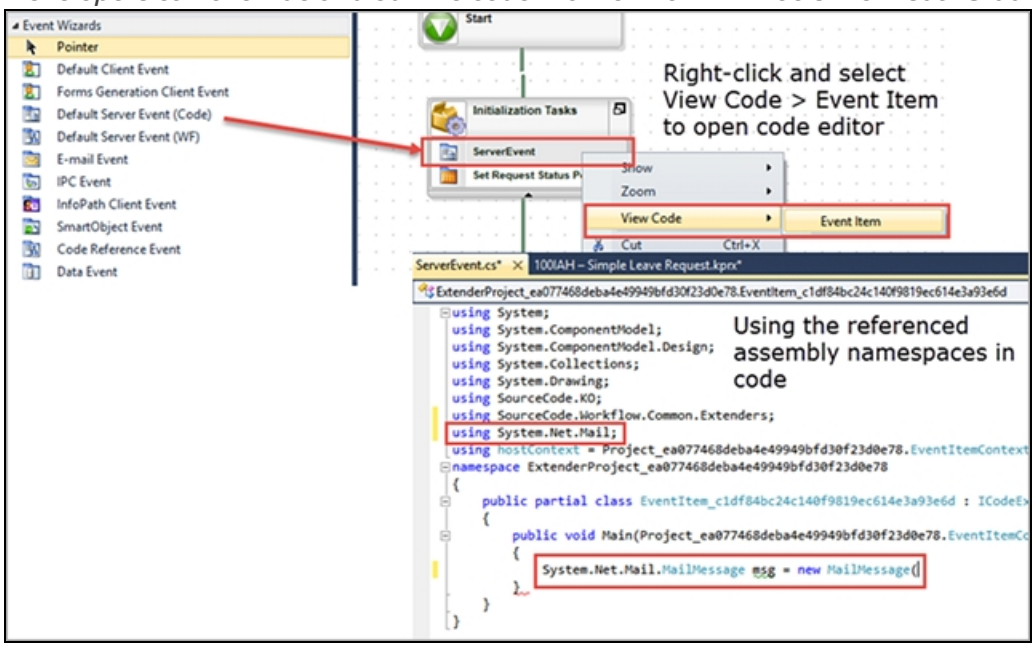
An example of a more advanced outcome that evaluates business data



Workflows can use SmartObjects and Wizards to integrate with external systems, such as updating a database, creating a new user account in Active Directory or sending an email. Workflow statistical, auditing and reporting data is automatically tracked and stored by K2 indefinitely, which provides a valuable tool for organizations to keep audit trails for their business processes.

Although much of K2 development does not require code, it is possible for developers to use tools like Microsoft Visual Studio to "get into the weeds" and edit the code in a workflow.

Developers can override and edit the code in a workflow with tools like Visual Studio



Summary

- Build workflows in the browser or using tools like K2 Designer, K2 Studio and Microsoft Visual Studio.
- User Tasks and System Tasks work together to make up a workflow.

- Escalations such as sending a reminder 2 days before the due date or by a specific date. Multiple escalations on the same task are possible.
- Serial and Parallel execution, workflows can split and merge back later, tasks can be assigned in serial or parallel fashion. Other patterns like n-level approval and “spider” patterns are all possible.
- Assign tasks to groups of users and wait for one, multiple or all users to complete the task.
- Multiple users’ inputs on the same task, for example voting scenarios (at least 2 managers need to approve the expense claim if it is more than $2000).
- Integrate with systems through wizards and SmartObjects (easy, no-code integration based on wizards and SmartObjects).
- Automatic tracking of reporting metrics and task auditing such as time to complete tasks, how long workflows execute, what users do what actions on a task.
- Developers can write code or customize any wizard with custom code when using K2 for Visual Studio.

Mastery Checkpoint: Understanding K2 Applications and application elements

Video

This is a checkpoint for the information covered in Part 2 of this module: Understanding K2 applications and application elements. If you are attending a training session led by a K2 instructor, this is your opportunity to ask questions around K2 application components, SmartObjects, SmartForms and workflows.

These are the main concepts you should understand:

- K2 applications can consist of Forms, Workflows, Reports and Data
- SmartForms are one option for the Forms component
- SmartObjects are the primary mechanism for data access and integrating with other systems in K2
- SmartObjects are a “middle layer” that allow consumers of data (e.g. Forms, Reports and Workflows) to interact with providers of data (e.g. SQL databases, Active Directory, SharePoint)

Knowledge-check questions

To check your own knowledge and comprehension, consider the questions and answers below. Some are specific to K2, others may require you to think how K2 applies to your environment.

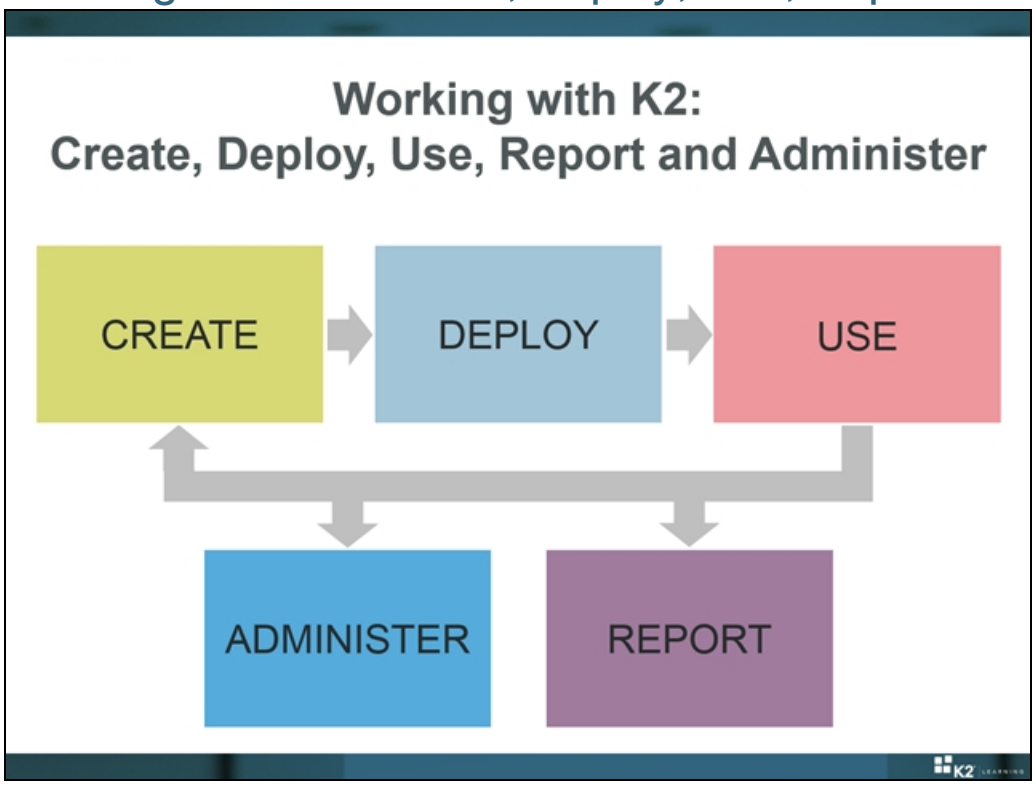
Q: What UI technology do you use in your organization? Do you plan to use SmartForms?
A: (no right/wrong answer, discussion question)

Q: What are SmartObjects?
Reveal answer
A: They are a “middle-layer” that allows consumers of data (e.g. Forms, Reports and Workflows) to interact with providers of data (e.g. SQL databases, Active Directory, SharePoint) a Data Access Component, an abstraction layer that simplifies data access in K2

Q: Can SmartObjects be used by non-K2 applications? Such as a ASP.NET web application?
Reveal answer
A: Yes, as long as those applications use the relevant K2 APIs/Services to interact with SmartObjects

Q: SmartForms are the only user interface technology you can use when working with K2
Reveal answer
A: False. SmartForms are just one UI technology option, you can use other UI technologies like ASP.NET forms as well

Working with K2: Create, Deploy, Use, Report and Administer



Video

In Part 3 of this module, we will briefly describe the tools and interfaces used to work with K2. The best way to do so is to break down the process of working with K2 into high-level "stages": Create, Deploy, Use, Report and Administer, and in the next few topics we will explain what tools are used during each of these stages.

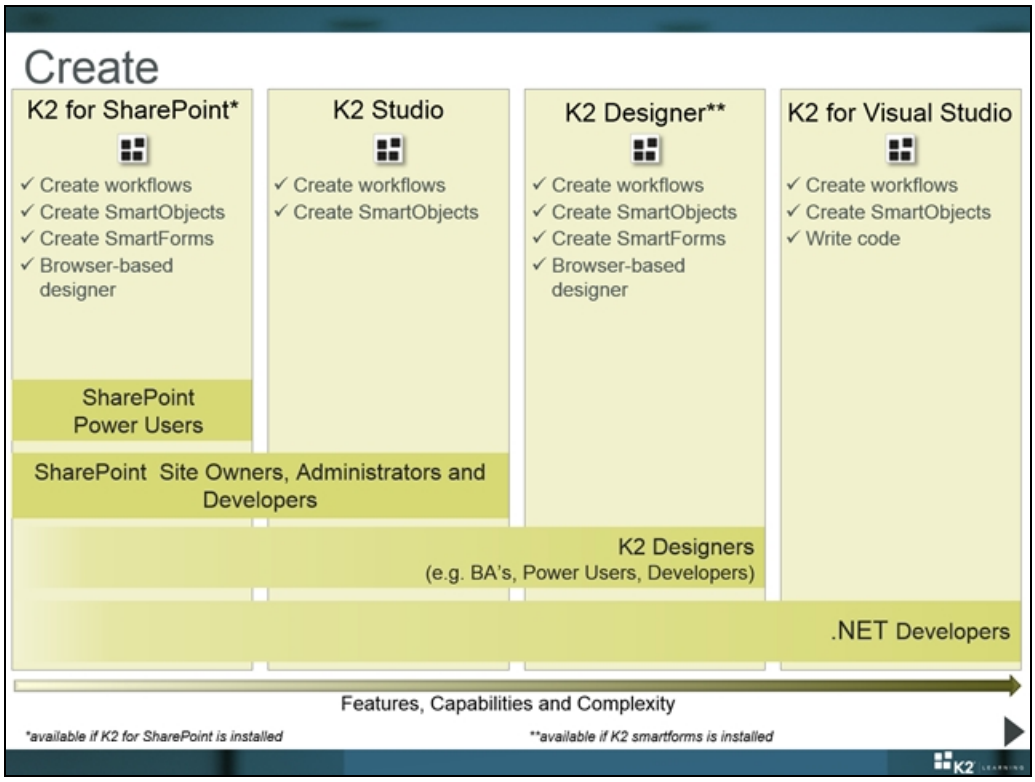
In the create stage, developers and designers build the actual applications and application elements in the appropriate design tool. Once the application is created, the application elements are deployed or “published” to an environment, either directly from the design tool or by using tools like K2 Package and Deployment to move applications between environments.

Once the application is deployed, end users interact with the applications we create, such as completing tasks, opening forms, starting workflows. While the application is being used, K2 will automatically track workflow metrics which can be used to run standard reports against workflow reporting data, or build and run custom reports that combine workflow reporting data with business data. Finally, administrators can maintain, monitor and administer the K2 environment by managing the running workflows, tasks, repairing errors, managing permissions and so forth.

Tip
With business dynamics and requirements constantly changing, you can expect to rework or change a K2 application multiple times over the course of its lifespan; this is not unusual and in fact indicates that the application is being used and needs to respond to the needs of the business. Also, you may want to implement a subset of functionality in the first version of an application and then add to the functionality with subsequent versions of the application.
It is common for K2 applications to go through several iterations due to enhancements or changing business requirements, and it is possible to update existing applications with newer versions when needed, which is why

the diagram "loops back" to the Create step.

Create



Video

During the Create stage, designers and developers use K2 tools to build applications with Forms, SmartObjects, Workflows and perhaps Reports. The K2 platform provides several design tools, each of which is targeted at particular roles or requirements in an organization. All of the artifacts created in the design tools are deployed to and executed on the same K2 server environment, and share the same robust enterprise-level capabilities of the K2 host server platform.

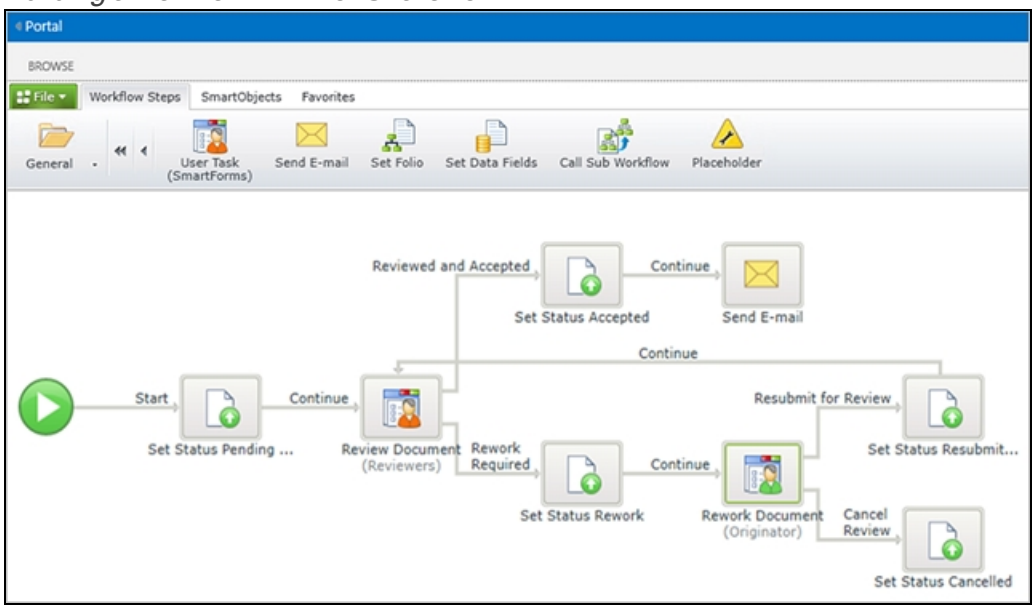
The designed artifacts share a common language, so it is certainly possible to work collaboratively and move a K2 process between designers, or to re-use artifacts created in one designer in another designer. For example, Business Analysts may create a K2 process using K2 Studio, and then hand it off to a developer to write code in the process using Visual Studio. The developer can then hand the process back to the business analyst who can then continue making changes to the process design. Another example is that a developer can create a SmartObject in K2 Studio, and Business Analysts can then use that same SmartObject in the web-based K2 Designer to create a user interface.

K2 for SharePoint

This design tool is hosted in a SharePoint site and is intended for SharePoint super users, site admins and farm admins to create SharePoint-centric workflows or applications that are used in the context of a SharePoint Site. K2 for SharePoint is typically used to build Applications that reside in SharePoint (such as SharePoint Workflow Applications and Business Applications that combine SharePoint Data with external data, along with customized User Interfaces (Forms), Workflows and Reports).

No K2 components need to be installed on the user's computer: K2 for SharePoint allows SharePoint users to create SharePoint Item-centric K2 applications using only a web browser, but with all of the power and flexibility of K2's integration and workflow features.

Building a workflow in K2 for SharePoint



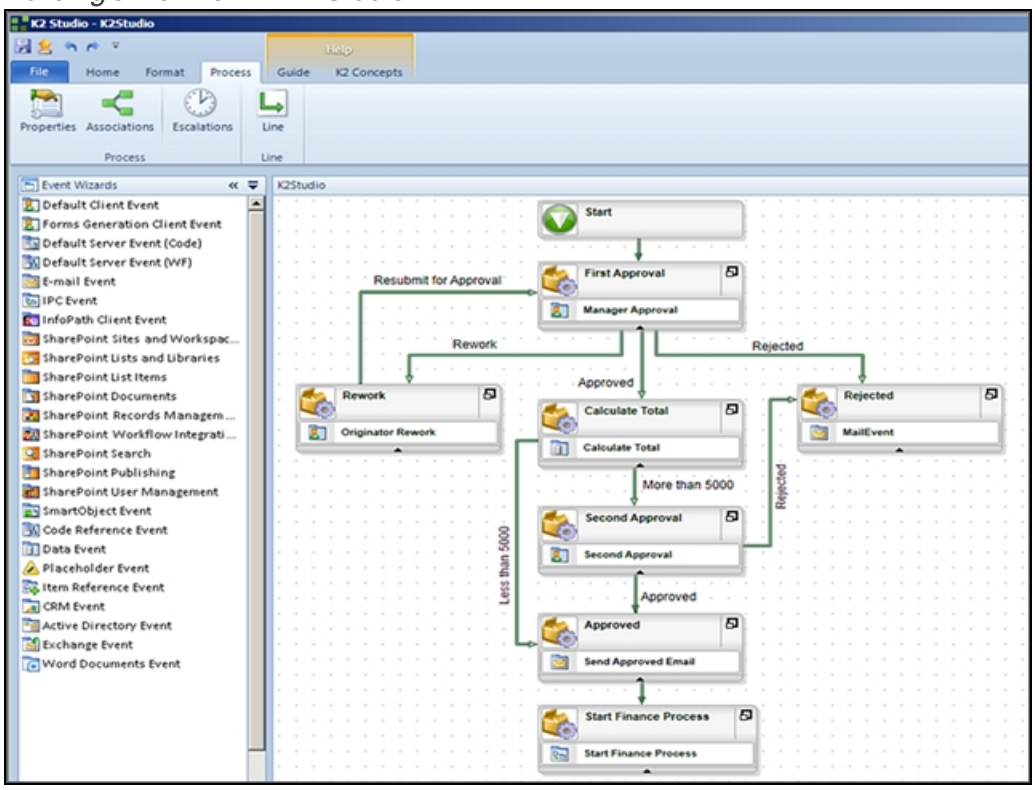
K2 Studio

K2 Studio is a visual design environment which is targeted at more technical roles like SharePoint site owners, administrators and SharePoint developers, as well as Business Analysts and workflow developers. This design tool must be installed on the user's computer, and allows process designers to assemble K2 workflows and create K2 SmartObjects.

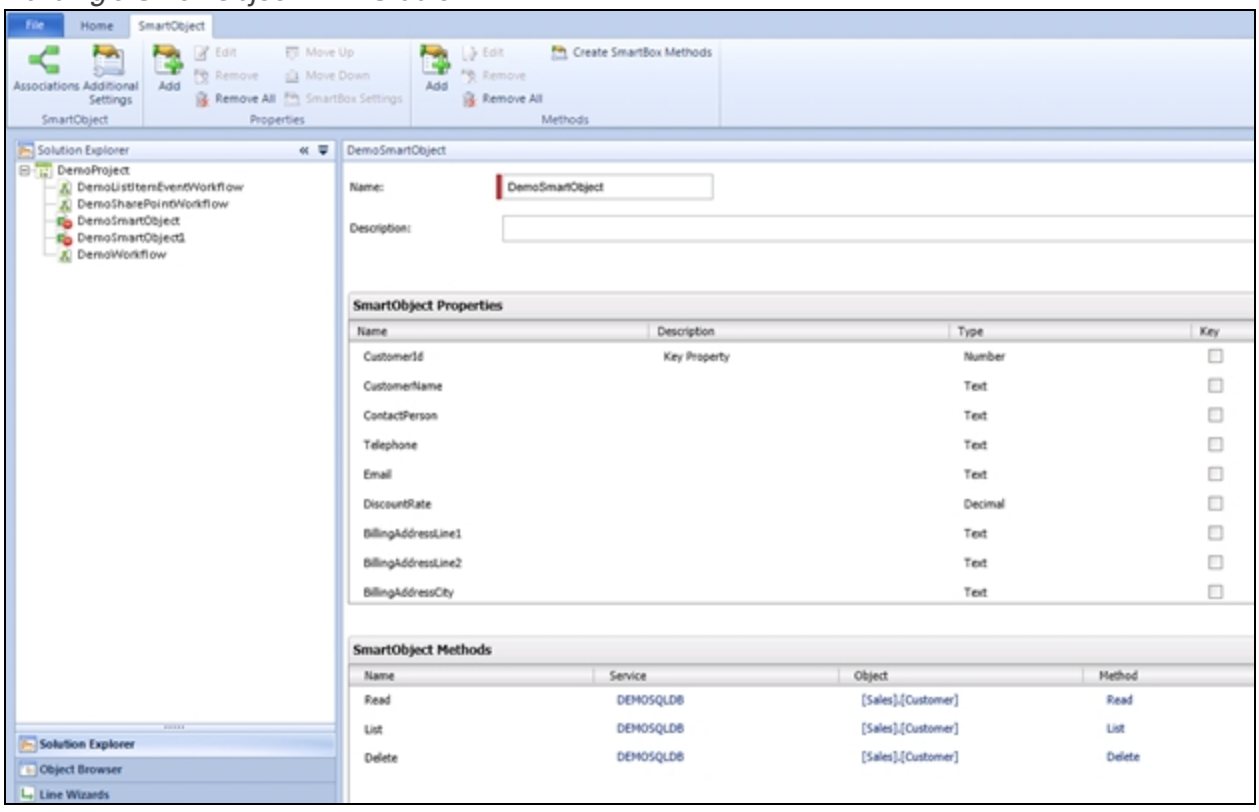
K2 Studio can be used to create everything from simple to advanced workflows using the full set of K2 wizards and integration points, as well as simple or advanced SmartObjects that integrate with external systems.

K2 Studio has a few more features than the K2 Designer for SharePoint, can be used to create processes that are not SharePoint-centric and can be used to create very complex and advanced processes. Designers can create free-form workflow layouts with features like multiple events per activity, multiple complex outcomes, complex task routing and repeating steps. Effectively, K2 Studio is the "next step" in complexity and allows designers to create everything from simple to advanced workflow designs, using generated Forms, InfoPath Forms, existing K2 SmartForms or custom user interfaces. The only restriction in K2 Studio is that it is not possible to write custom .NET code in this design tool: to do this, you will need to use K2 for Visual Studio.

Building a workflow in K2 Studio



Building a SmartObject in K2 Studio

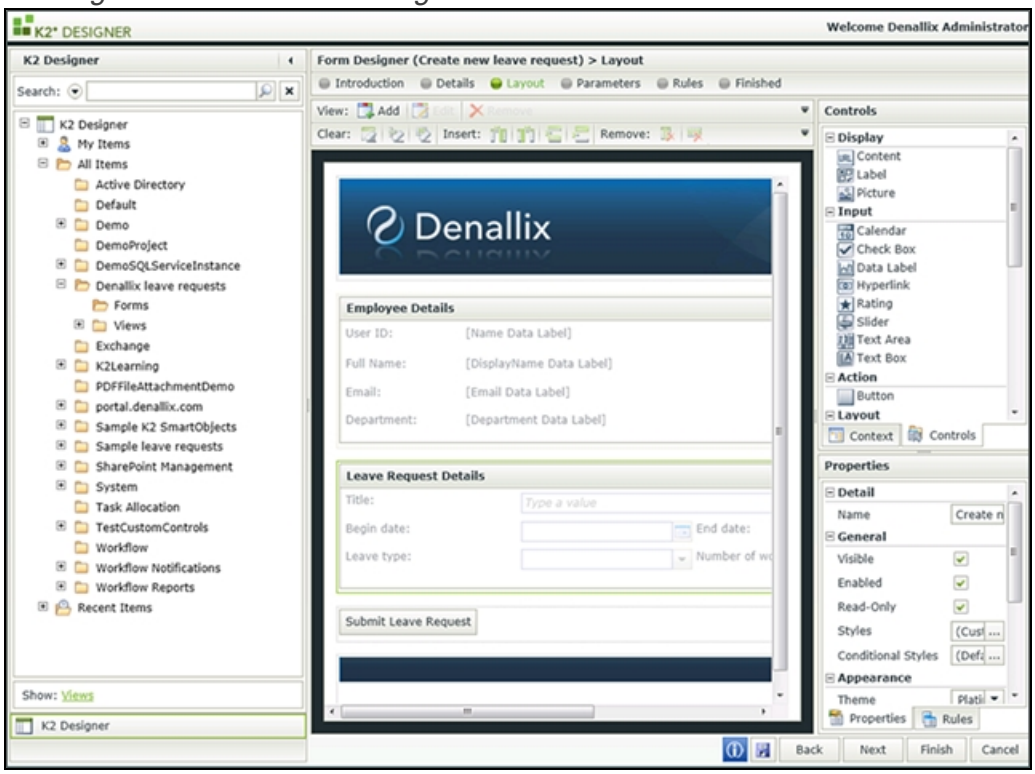


K2 Designer

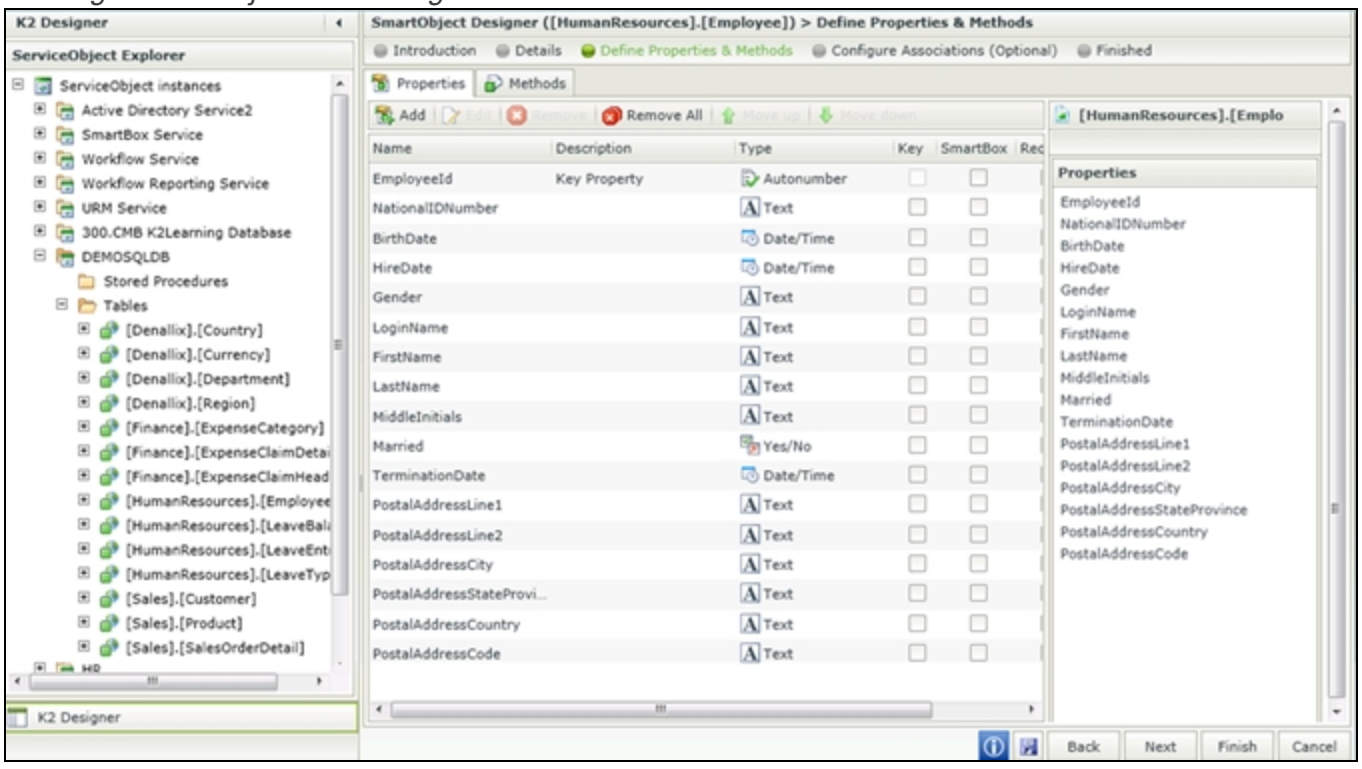
The K2 Designer is a web-based design environment which allows K2 Designers and Developers to build workflows, SmartObjects and SmartForms using wizards and a drag-and-drop interface. K2 Designer does not require any client-side installation. Designers can use most modern web browsers to build complete solutions in this design environment, and all artifacts are stored on the K2 server.

Note
K2 Designer is only available if K2 smartforms has been installed on the K2 environment and is currently the only design tool that is used to create SmartForms.

Building a SmartForm in K2 Designer



Building a SmartObject in K2 Designer



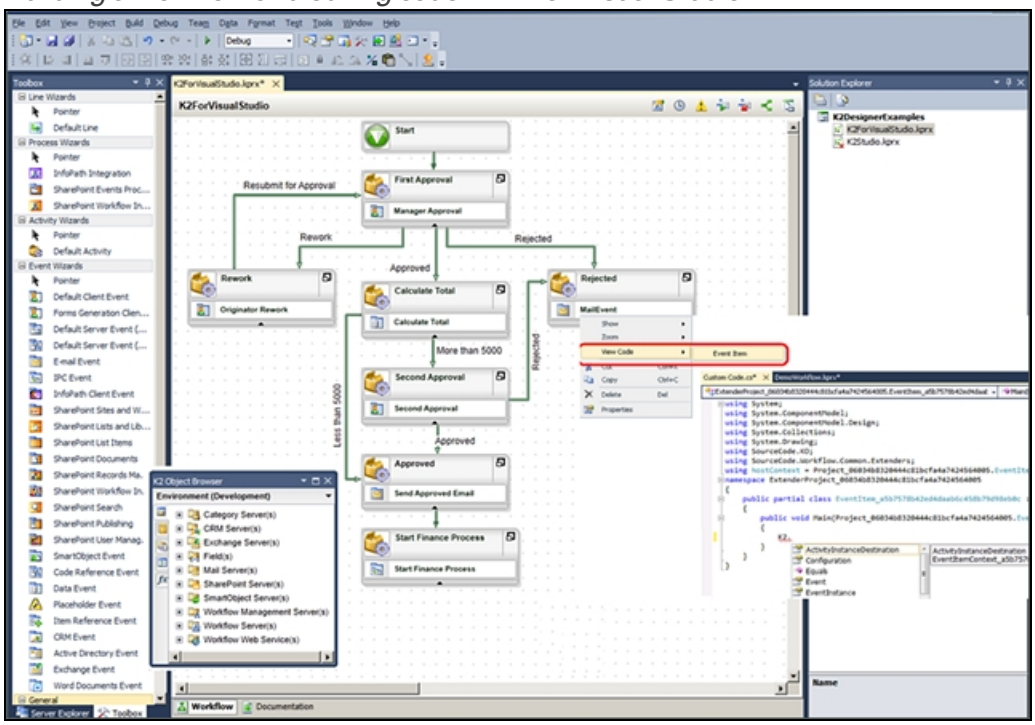
K2 for Visual Studio

The K2 for Visual Studio is intended for .NET developers and can be used to build workflows and SmartObjects. This is the most powerful and flexible design environment, and allows developers to write custom .NET code in workflows and

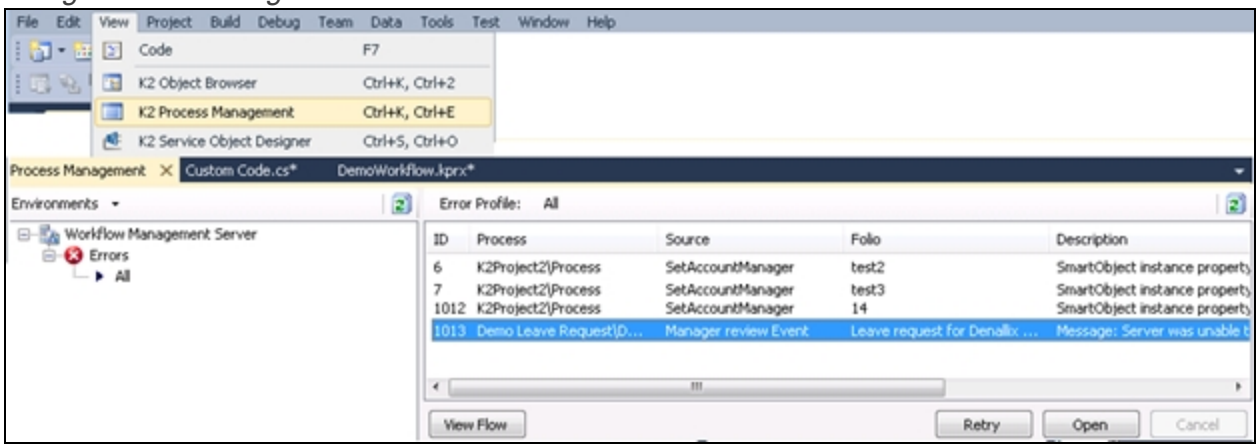
leverage other Visual Studio capabilities like built-in integration with Source Control providers while still being able to use the standard K2 wizard-driven drag-and-drop design experience. This design environment has the same wizards and workflow design capabilities as K2 Studio and it is possible to open and edit the same workflow in either of the designers, making collaborative process design possible.

K2 for Visual Studio also allows developers to debug workflows, repair errors and migrate processes in error state from one version of a workflow to a newer version. Further, developers can drill-down and override the behavior of any of the wizard-based events in the workflow, allowing them to extend or modify the workflow execution as required.

Building a workflow and editing code in K2 for Visual Studio



Using Process Management in K2 for Visual Studio

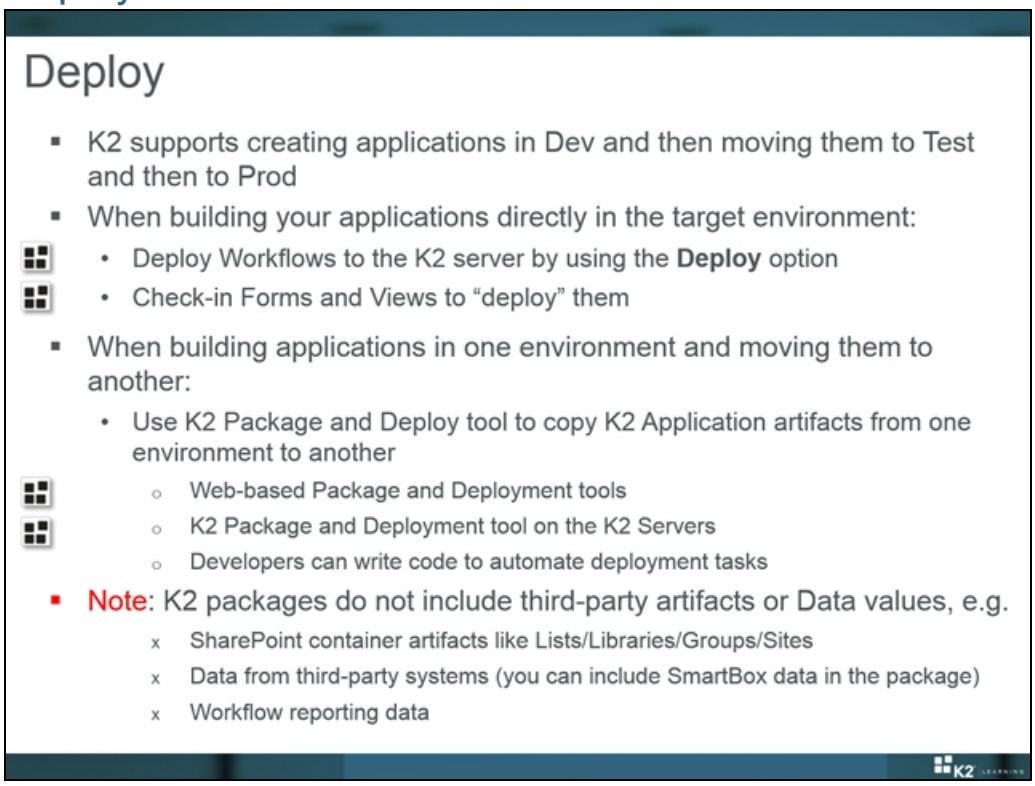


Summary

- K2 provides several different design tools depending on the project's scope and the roles that will be implementing the solution.
- Some K2 projects are built by non-developers using only web-based design tools, while other projects may be built by developers who need to write code using K2 design tools in Microsoft Visual Studio.
- Capabilities increase as the tools get more complex, but fundamentally they all share the same DNA and produce output in the same format, deploying to the same execution engine.
- The design tools are just differently-scoped and have different capabilities, and the complexity and capabilities GENERALLY increase as shown in the slides above.

- The audiences and capabilities listed in the slides are guidelines and suggestions, of course each project may have its own particular requirements.
- Generally speaking (with a few exceptions), it is possible to move a solution's elements from one designer to another, for example starting with a workflow in K2 Studio and then editing it further in Visual Studio to add some code.

Deploy



Video

You must deploy your K2 Application elements before they can be used. The method you use to deploy your applications depends on whether you are building the application in the same environment where it will be used, or if you want to create an application in one environment and then package and deploy the application to a different environment or site. When you are ready to publish your solution, deployment can be as easy as selecting a different target environment and publishing the artifacts directly from the design tool you are using, or packaging up a solutions and handing the deployment package to a system administrator for deployment to another environment. K2 supports the use of multiple environments (for example, DEV, TEST and PROD) and depending on the organizations change management policies, you can configure K2 differently to allow or prevent users from deploying applications to Production.

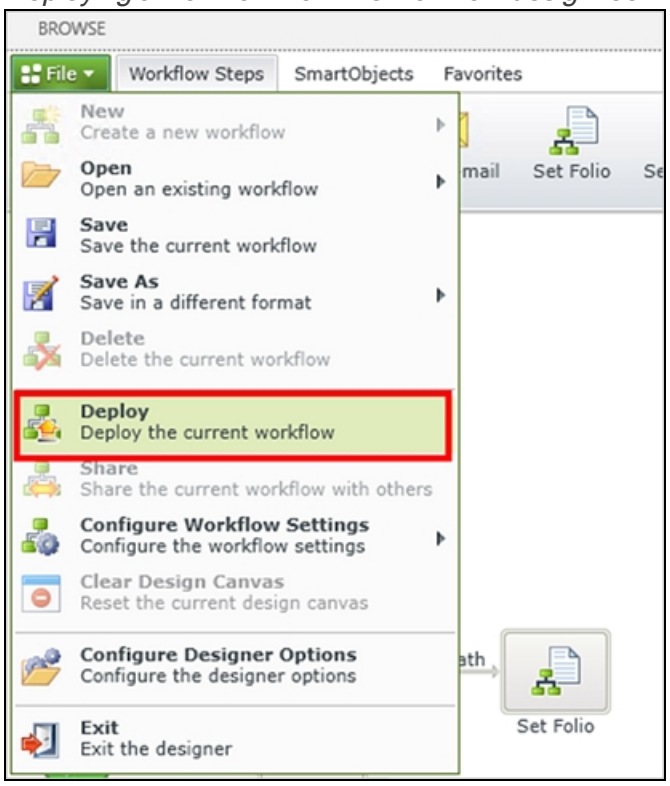
It is not necessary to re-build a solution so that it will work in the target environment. Of course, you should take care to insert placeholders for server names and so on. Keep in mind, in other learning modules, we will describe in more detail, some techniques and approaches you can use to make your solution deployable in multiple environments.

Note
By default, K2 will use the most recent workflow version that was deployed as the version for all new instances of the workflow, but any existing running instances of the workflow are left on the version that they were started on. K2 does not automatically "upgrade" any existing workflow instances to newer versions because you may have made significant changes to the workflow design between versions.

Deploying applications directly from the design tool

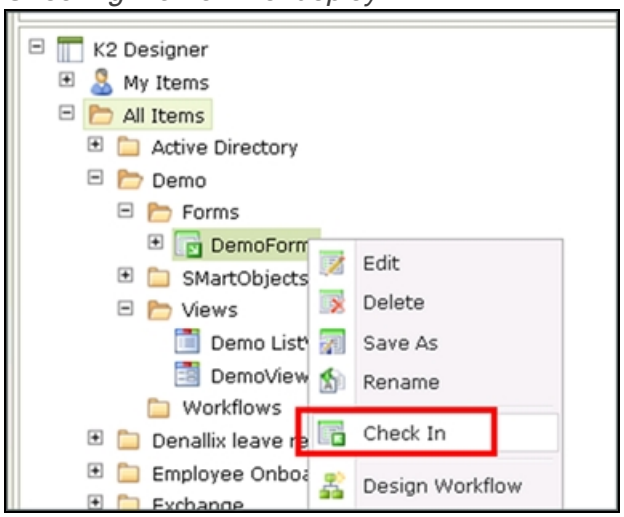
If your organization allows you to deploy directly to the target environment from the design tool, you can publish workflows using the **Deploy** menu option. In this option, you can select the environment where you want to deploy the workflow.

Deploying a workflow from the workflow design tool



If you need to make a change to a Form or View, simply check it out, make the change and then check it back in. Changes made to Forms that are checked out are not available to your users until checked back in. Changes to workflows require redeployment of the workflow before the change is available to your users.

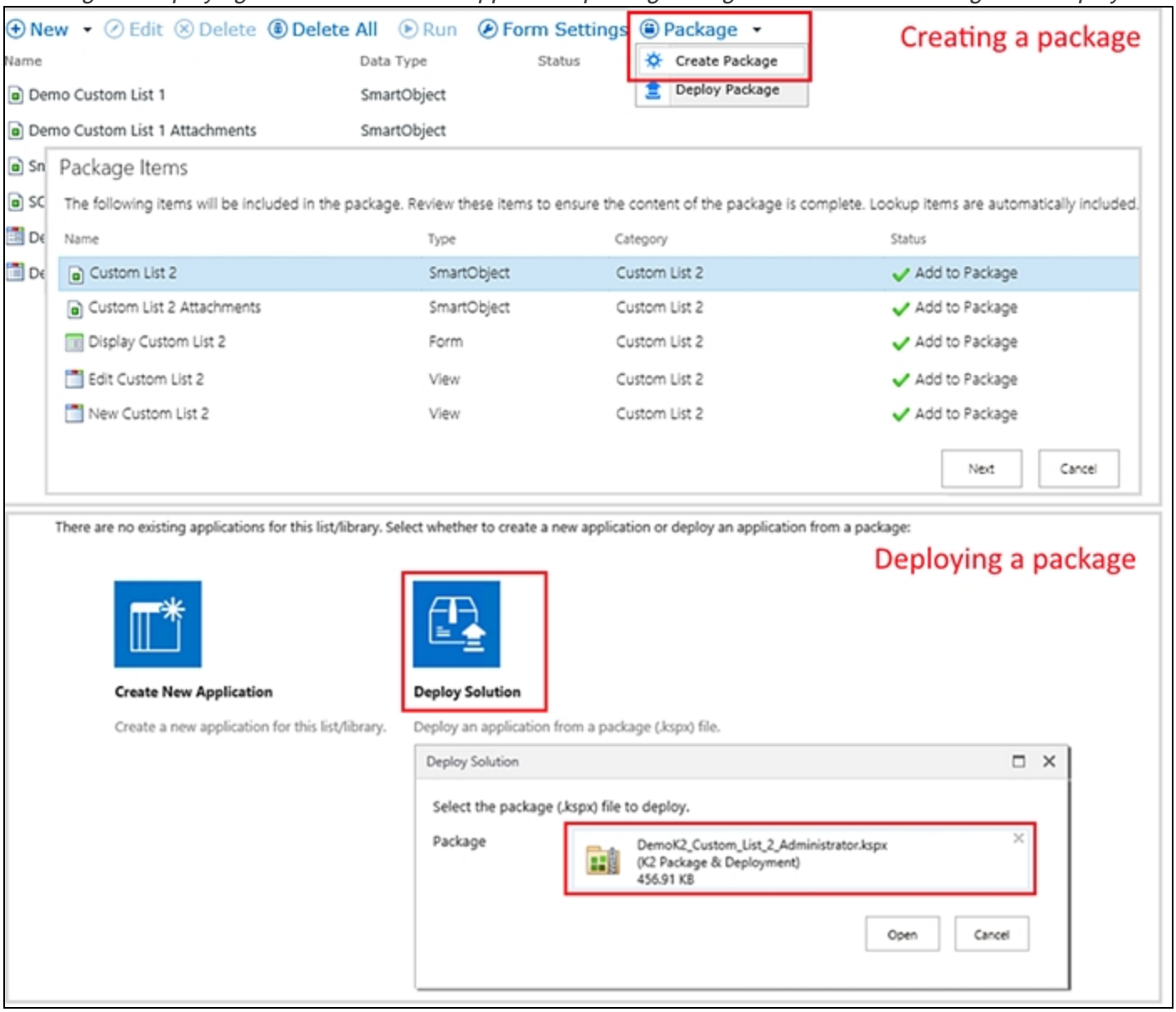
Checking in a Form to "deploy" it



Using Package and Deployment to deploy applications to another environment

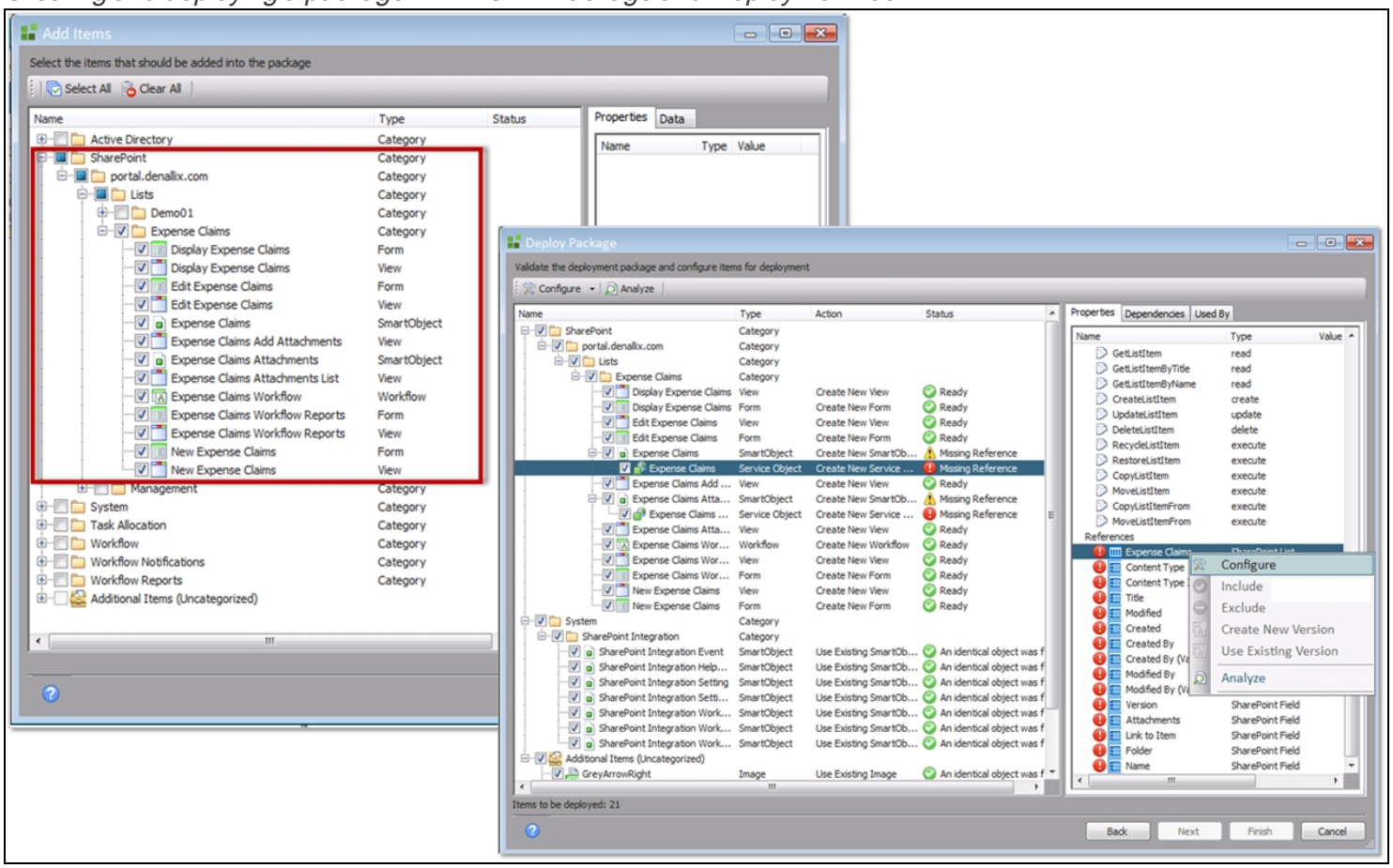
K2's Package and Deployment tool allows you to bundle all of your K2 artifacts (Forms, Views, Workflows, SmartObjects and the underlying Service Instances that the SmartObjects rely upon) into a package and then deploy it to another environment. If your organization does not allow you deploy applications directly to a target environment from a design tool, you will need to use the K2 Package and Deployment tool to create a package of the application elements in the source environment. Typically, a user with sufficient rights in the target environment will then use the same tool to deploy these application elements.

Creating and deploying a K2 for SharePoint application package using the web-based Package and Deployment tool



You may also use the thick-client, Microsoft Management Console (MMC)-based application to package and deploy applications. By default, this application is installed on the K2 server.

Creating and deploying a package with the K2 Package and Deployment tool

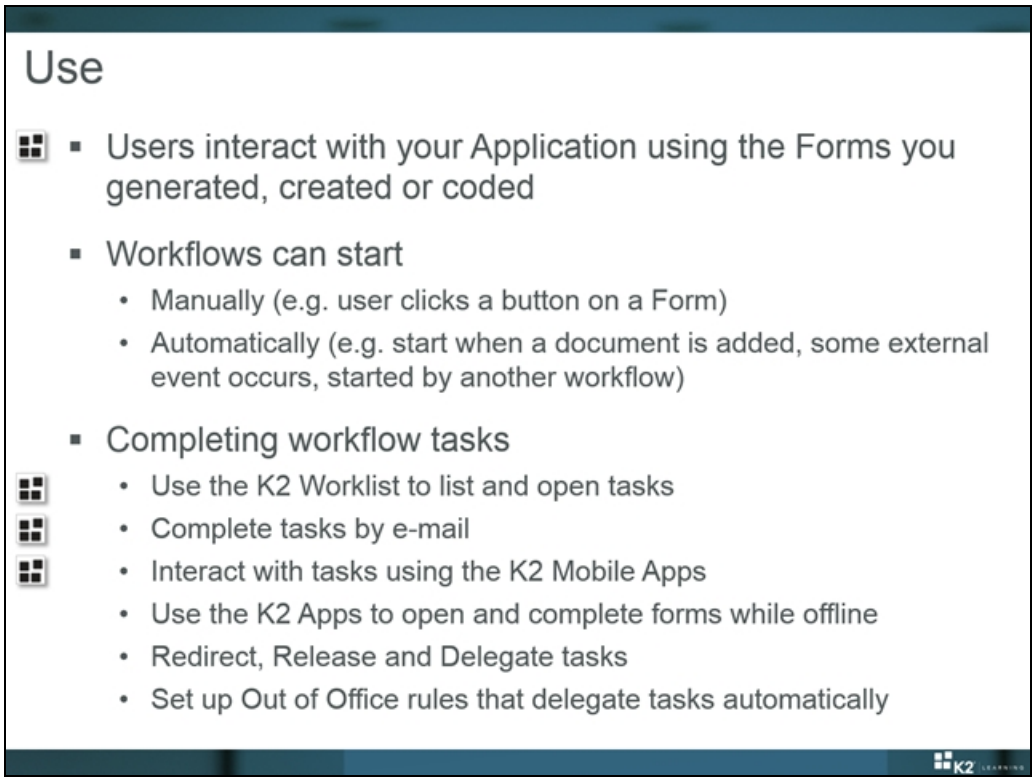


Note
K2 Package and Deployment only packages K2 application elements. It does not package third-party pieces like SharePoint artifacts (sites, lists and libraries), nor does it package data other than optionally including SmartBox-based data, and it will not package SQL databases or scripts to create these DBs.

Summary

- Application elements can be deployed directly from the design tools to a linked environment, or you can use tools like K2 Package and Deployment to deploy applications between different environments.
- You can use the web-based package and deployment tools or the thick-client Microsoft Management Console app to package and deploy K2 applications.
- K2 Package and Deployment ONLY packages K2 application elements.

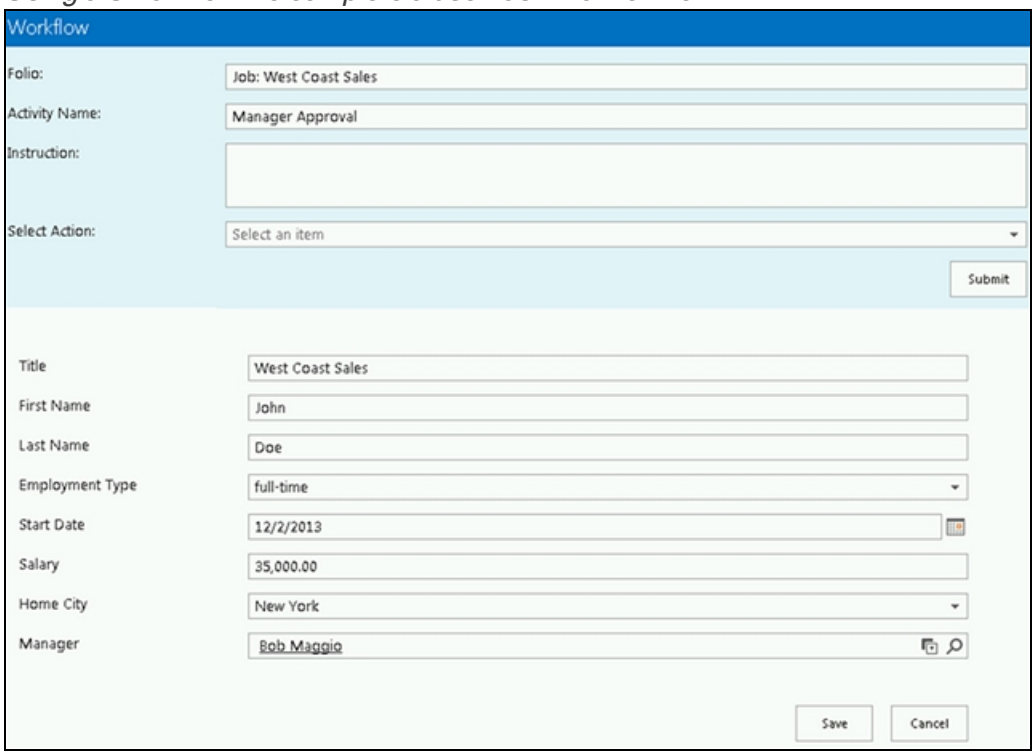
Use



Video

After deployment, users can begin using the application by working with the Forms you have created, performing an action that starts a workflow or completing their workflow tasks using emails or the K2 worklist. There are many different ways users can interact with your applications, it mostly depends on what you have decided to build into the application.

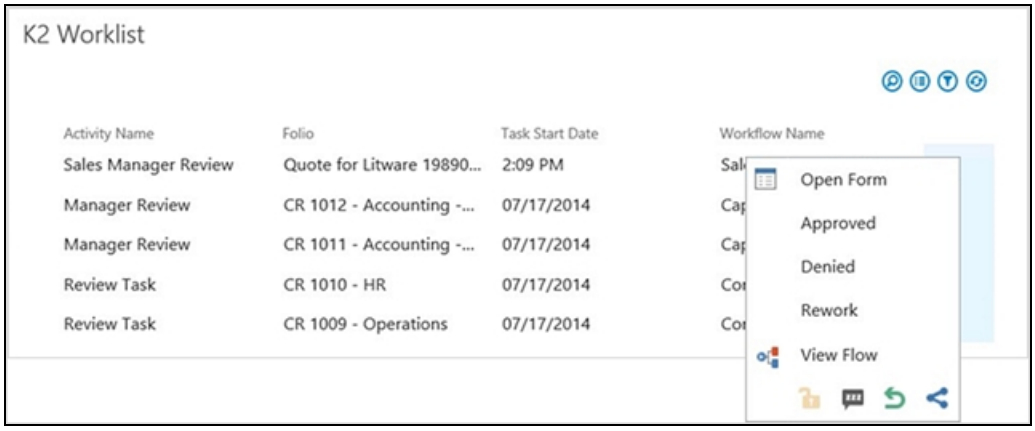
Using a SmartForm to complete a user task in a workflow



Depending on how the Start Rule for a workflow is configured, some workflows may start automatically when something happens (e.g. a new Item is added to a SharePoint List or it is started as a child workflow by another workflow), while others may start when a user clicks a button on a Form.

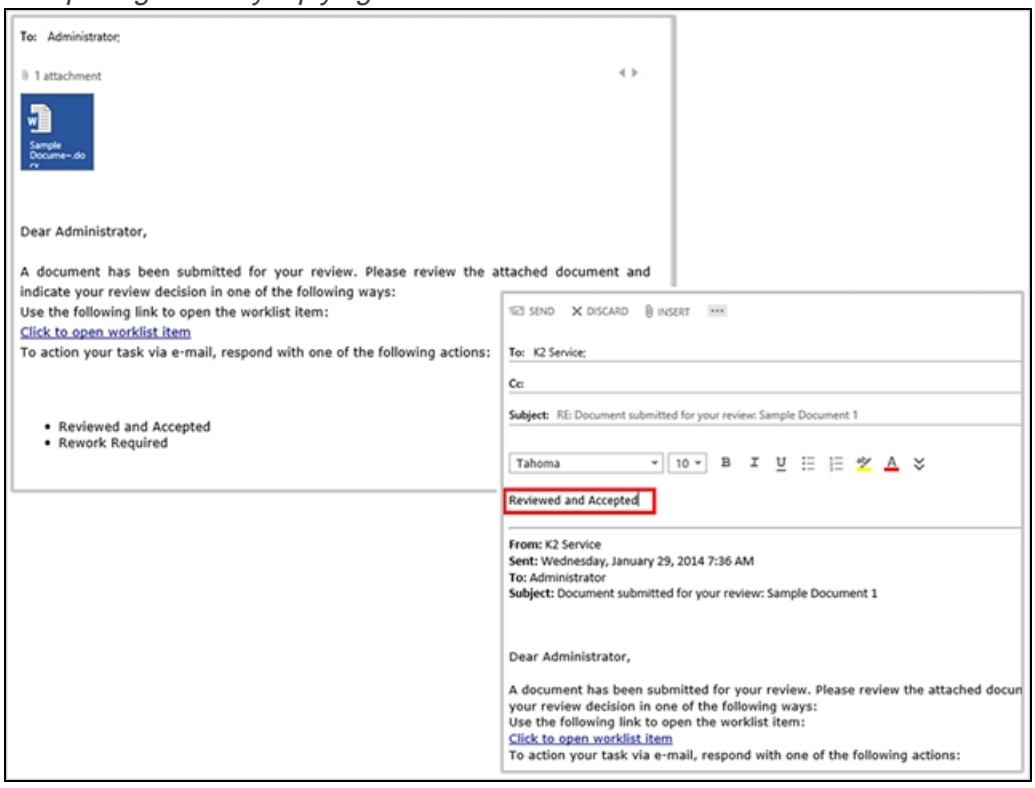
Users can complete their K2 workflow tasks in many different ways. The primary way in which users access their K2 tasks is through one of the K2 Worklist controls, but they can also be notified of new tasks by email.

The K2 Worklist



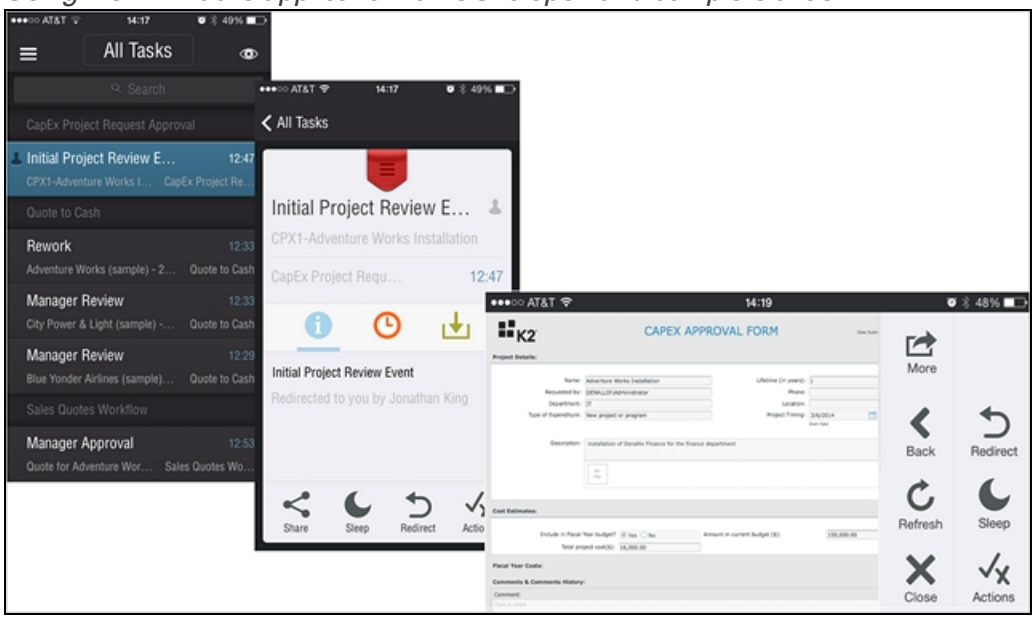
SmartActions is a feature in K2 which allows users to complete their tasks merely by replying to an email with one of the available decisions for that task. This is ideal in highly offline situations or when you want to give users a very simple way of completing tasks.

Completing a task by replying to the Task Notification email with a decision



Finally, users can also interact with tasks using one of the available K2 mobile applications. If the Form has been enabled for Offline Mode, the user can even work with the form while they are disconnected, and the app will automatically send the updated form data back to K2 when the device is online again.

Using the K2 mobile application for iOS to open and complete a task

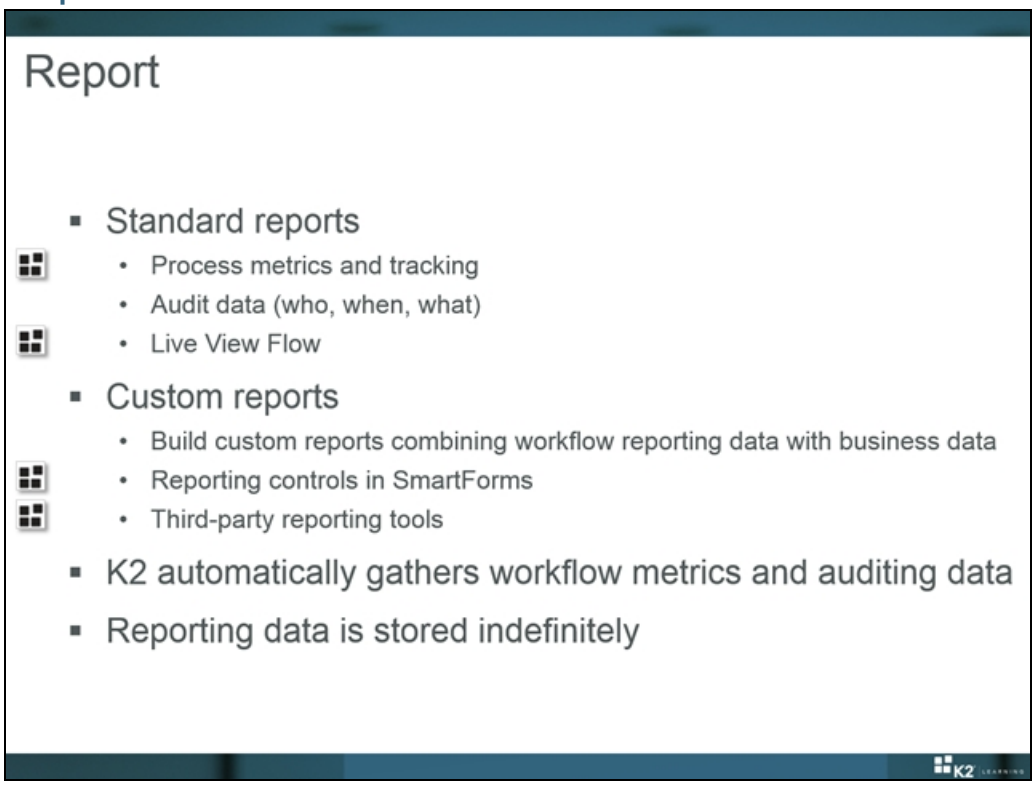


Users can optionally Redirect or Delegate tasks to other users, and they can also set up Out-of-Office routing rules that will automatically forward their tasks to other users while they are out of the office. This functionality is exposed on the K2 worklist.

Summary

- Workflows can start automatically (e.g. when a doc is uploaded) or manually (e.g. when the user clicks a button).
- Users open their K2 tasks using email notifications or the K2 worklist.
- Users can use the K2 mobile app to complete workflow tasks or complete tasks directly in email (if the workflow designer selected that option).
- Users can release, redirect or delegate tasks.
- Users can set up out-of-office rules to delegate tasks automatically during specific times.

Report



Video

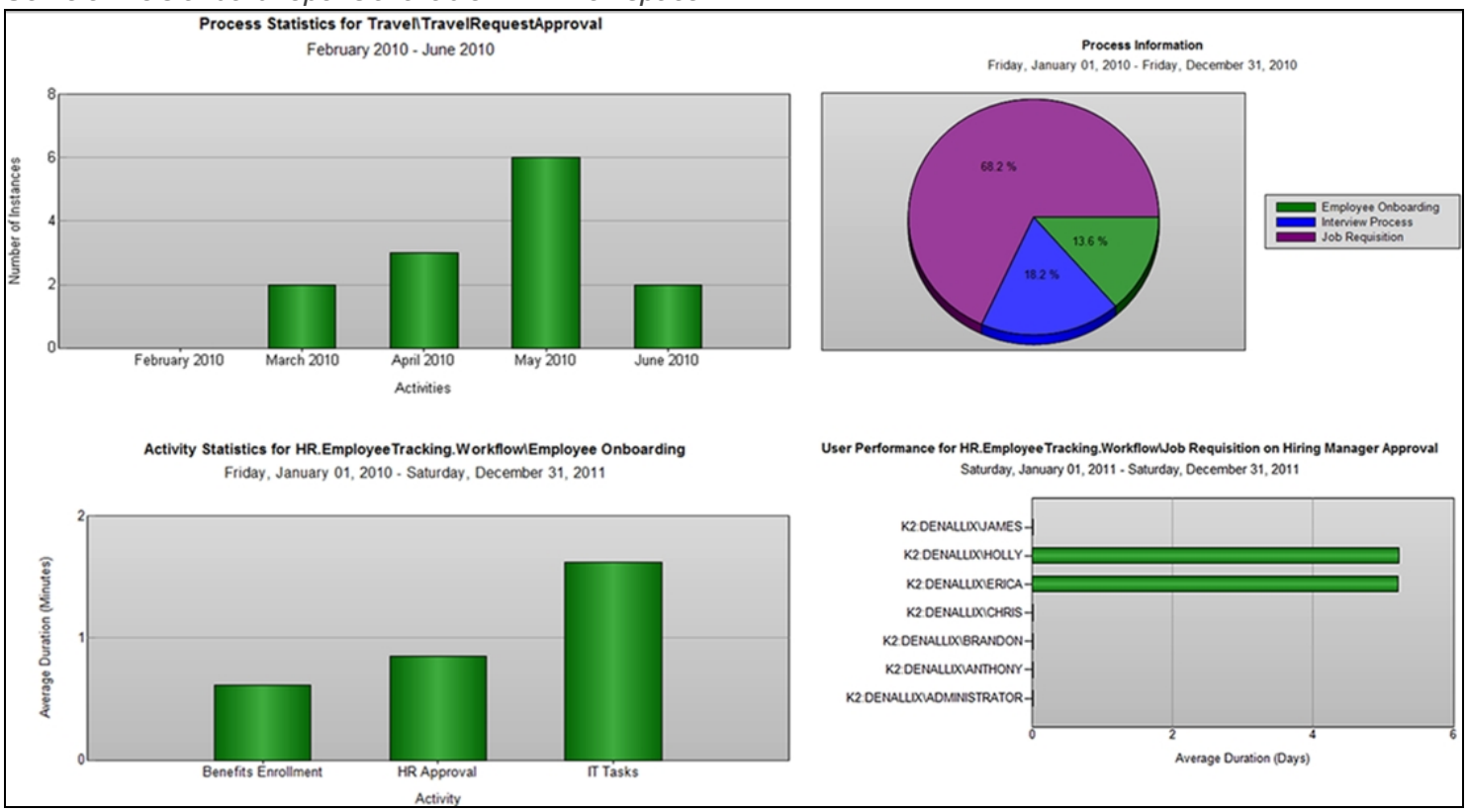
K2 offers several standard reports for workflows, both for active (running) workflows and workflows that have already completed. There are several interfaces available to report on and monitor K2 applications.

Note
K2 will automatically store workflow reporting data and metrics, and retains reporting history for workflows indefinitely.

K2 Workspace Reports

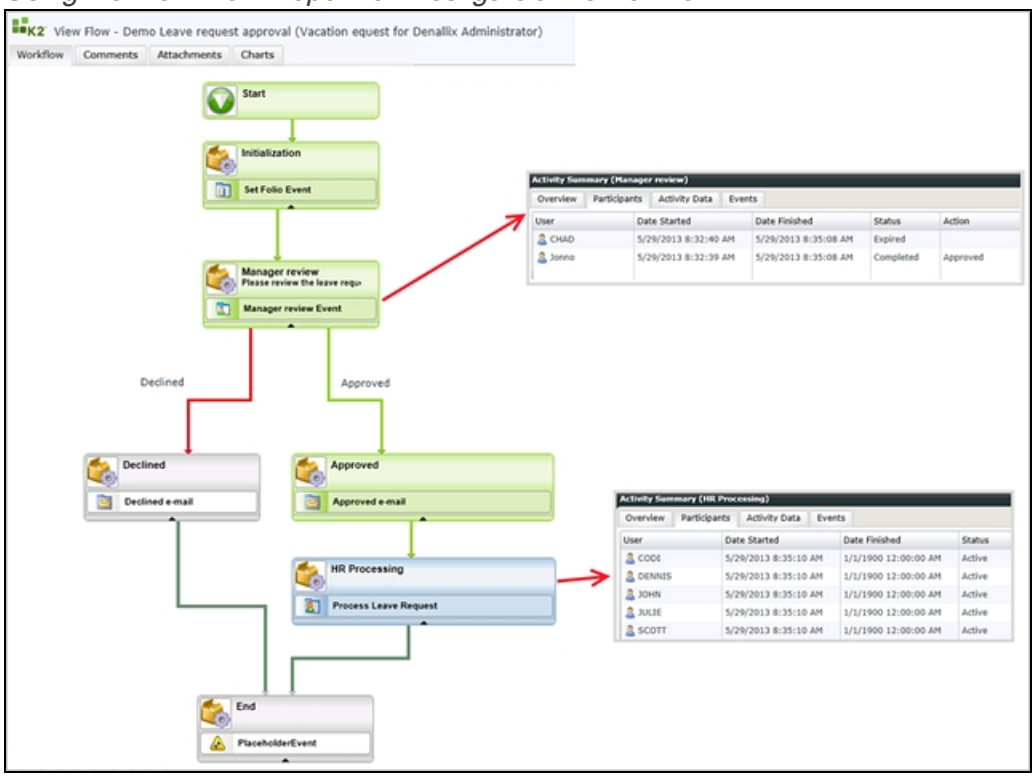
The K2 Workspace is a web-based interface that exposes K2 reports and administration tools. All the standard K2 workflow reports are available in the K2 Workspace. Authorized users can also use the custom report designer in K2 Workspace to build and run custom reports, combining both workflow reporting data and other business data into a single report.

Some of the standard reports available in K2 Workspace



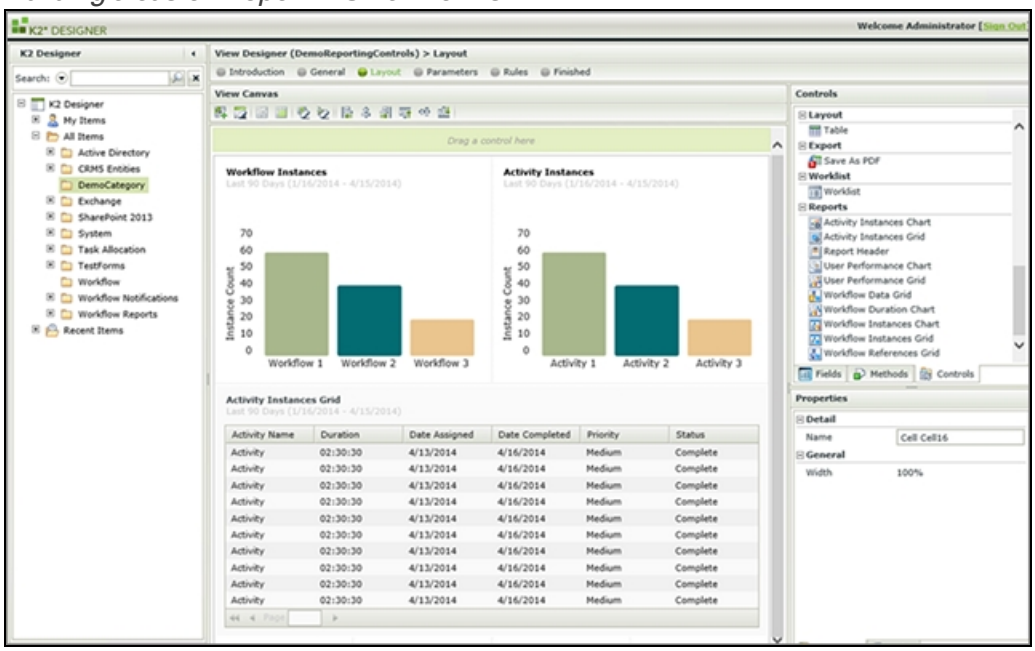
The View Flow report is a real-time reporting tool that can be used to query the state of active and completed workflows. It visually shows the path that a specific instance of a workflow took, and also allows you to drill down into specific steps of the workflow to see who actioned a task, when, and with what decision. This report can also determine (for an active workflow) where the instance is currently at, and which users need to complete the active task.

Using the View Flow Report to investigate a live workflow

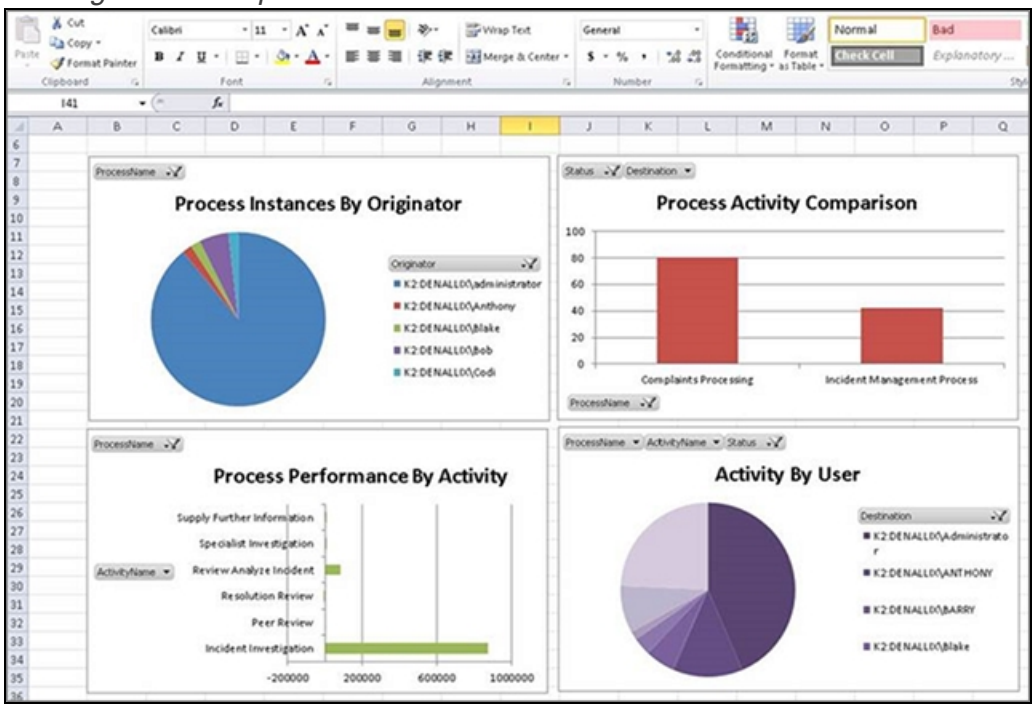


It is possible to create custom reports against workflow reporting data as well, either by using tools like the custom report designer in K2 workspace, the available reporting controls in K2 SmartForms or building reports in third-party tools that are able to interact with K2, such as Microsoft Excel or SQL Server Reporting Services.

Building a custom report in SmartForms



Building a custom report in Microsoft Excel



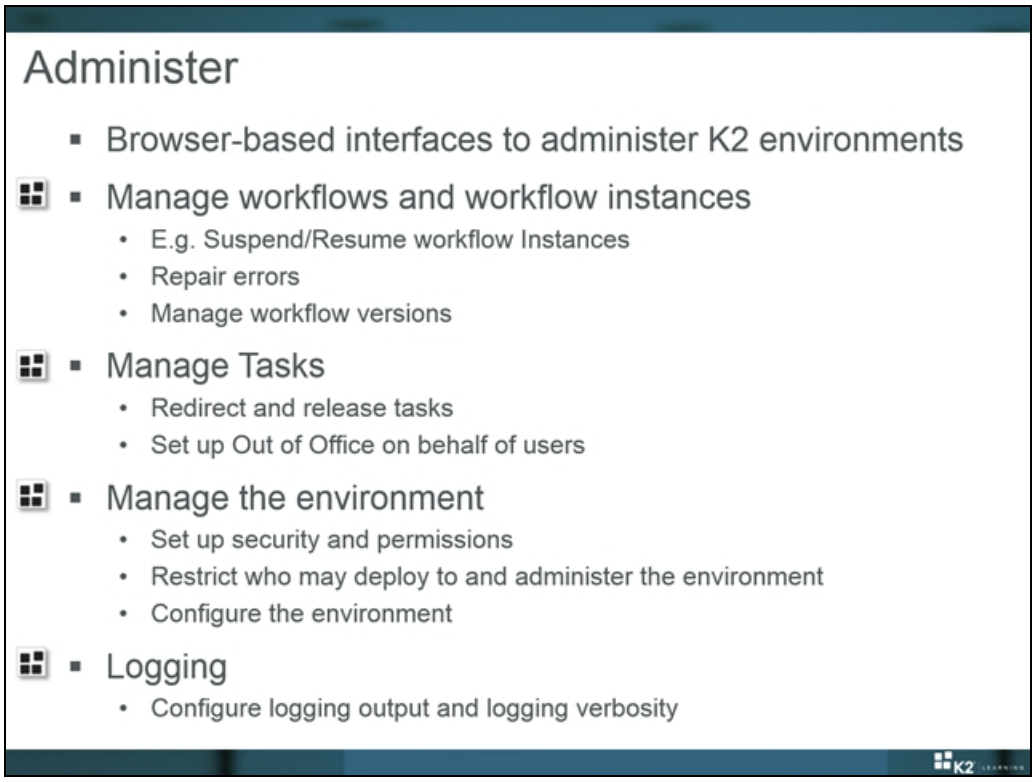
Summary

- You can access workflow reporting data through standard reports such as Process Information, User Performance, Activity Statistics and other reports.
- K2 provides a near-real time View Flow report which can display live process monitoring information, with drill-down capabilities.
- You can create custom reports to expose workflow data or combine workflow data with business data.
- Create custom reports in SmartForms using the SmartForms reporting controls.
- Create custom reports in K2 Workspace.

- Create custom reports by referencing the K2 ADO provider or the K2 web services in third-party tools like SSRS, Excel and PowerPivot to build completely custom reports.
- K2 gathers workflow reporting and auditing data automatically and stores the workflow reporting and auditing data indefinitely.

.

Administer



Video

K2 administrators may need to manage published processes or the K2 environment. Typical administration tasks include managing security on processes, performing manual override actions on active processes, redirecting work from one person to another, resolving error conditions, or administering the K2 server itself.

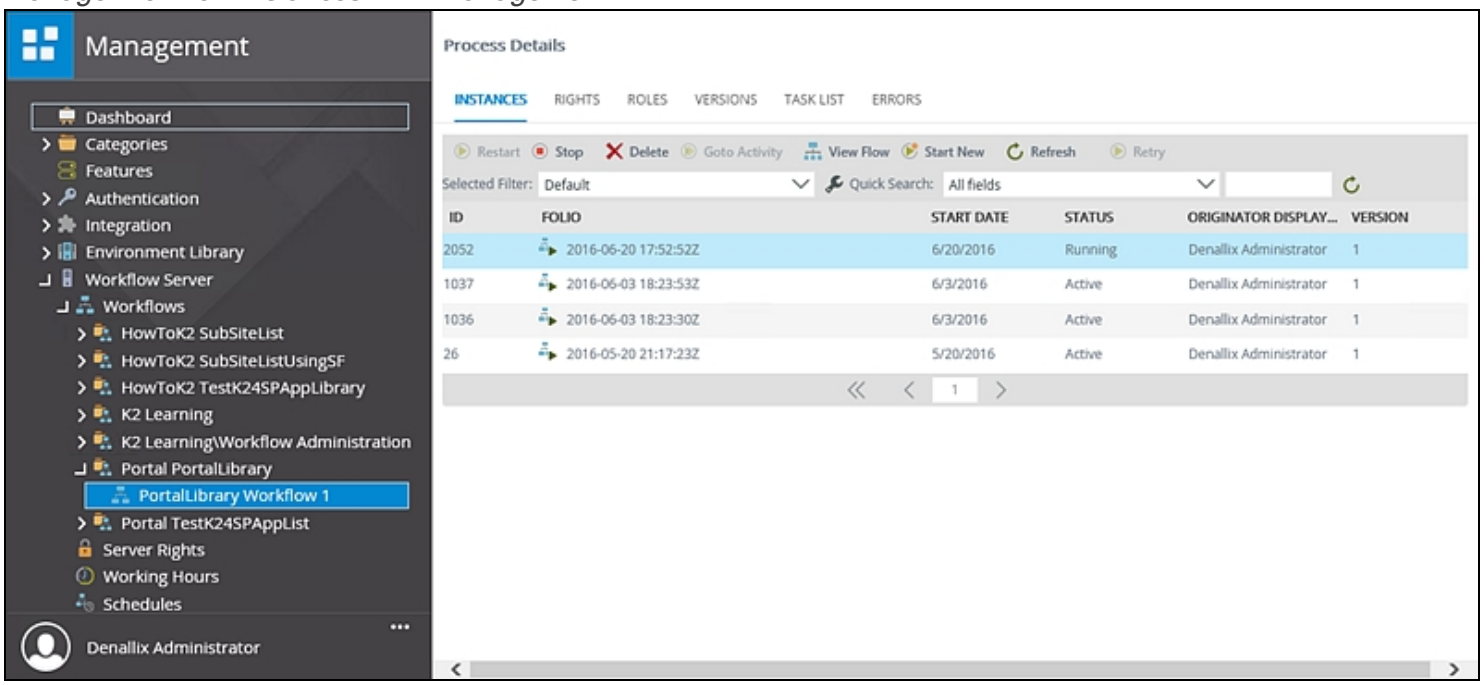
K2 provides web-based administration tools that allow authorized users to perform these administrative functions. Note that K2 will enforce security for these activities, and only users with administrative permissions on the K2 environment will be able to manage processes using the administration interfaces.

The K2 administration interfaces allow users to maintain security, repair workflows that have failed, maintain process versions by deleting or promoting different versions of the workflow, override task allocation, set up out-of-office task routing, and much more.

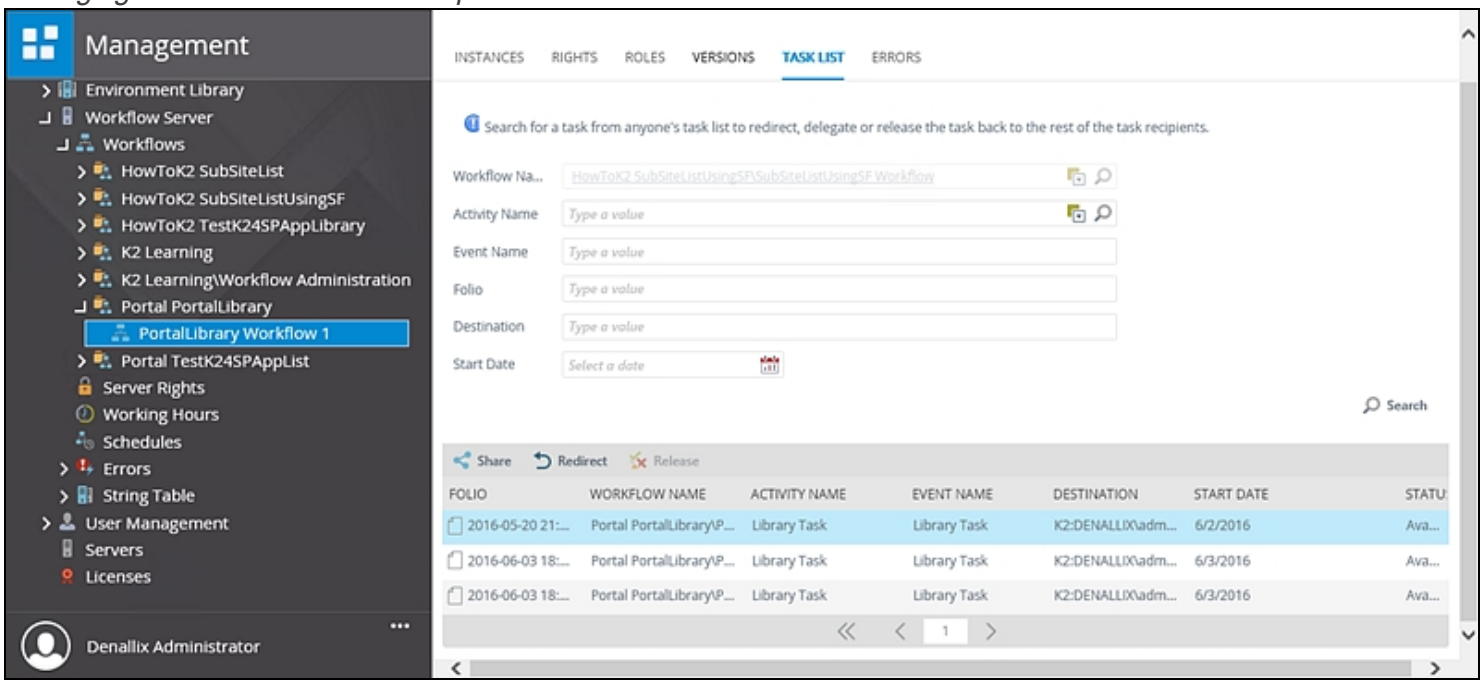
K2 Management allows process administrators and K2 administrators to manage process instances and the K2 server.

Note
A workflow instance is an instance of a particular workflow design (for example, Bob's leave request or Alice's leave request. Bob and Alice are two separate workflow instances of the same "Leave Request Approval" work-flow design.

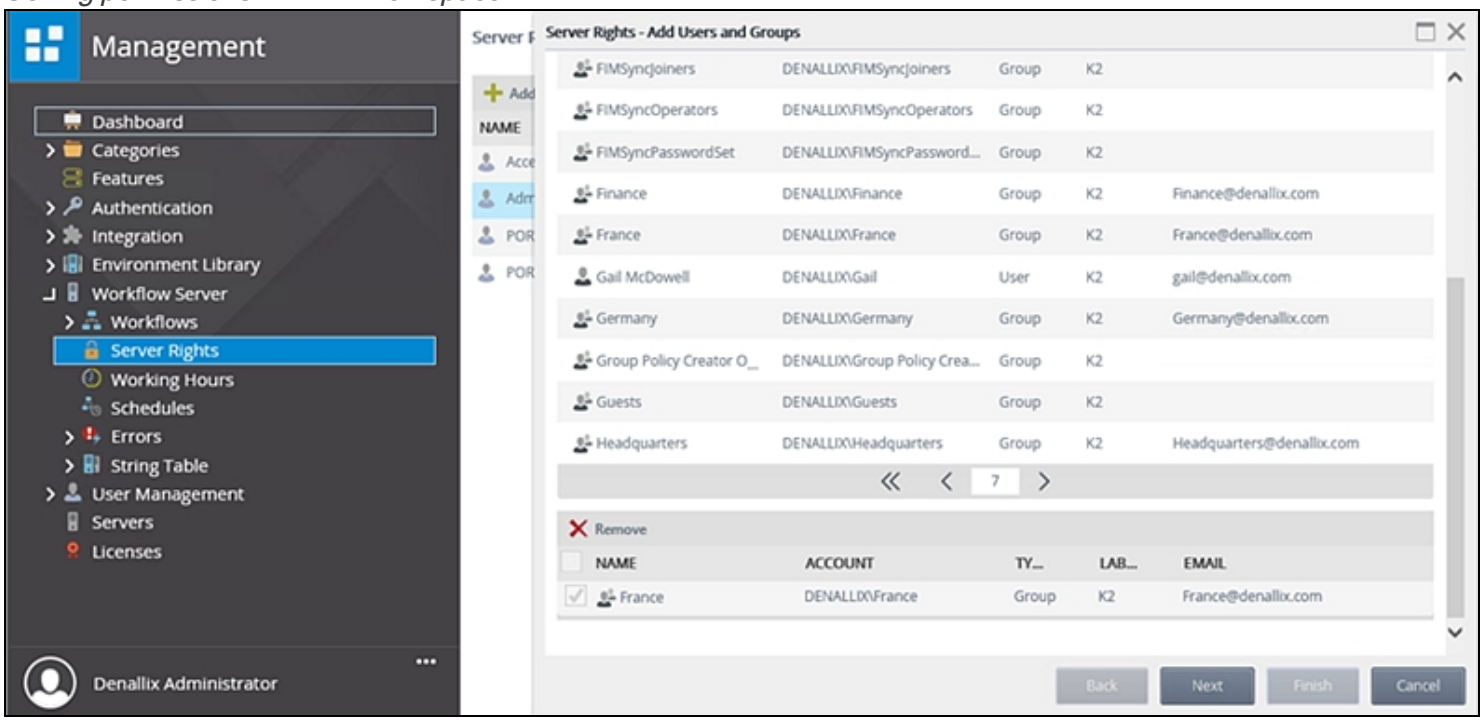
Manage Workflow Instances in K2 Management



Managing worklist tasks in K2 workspace

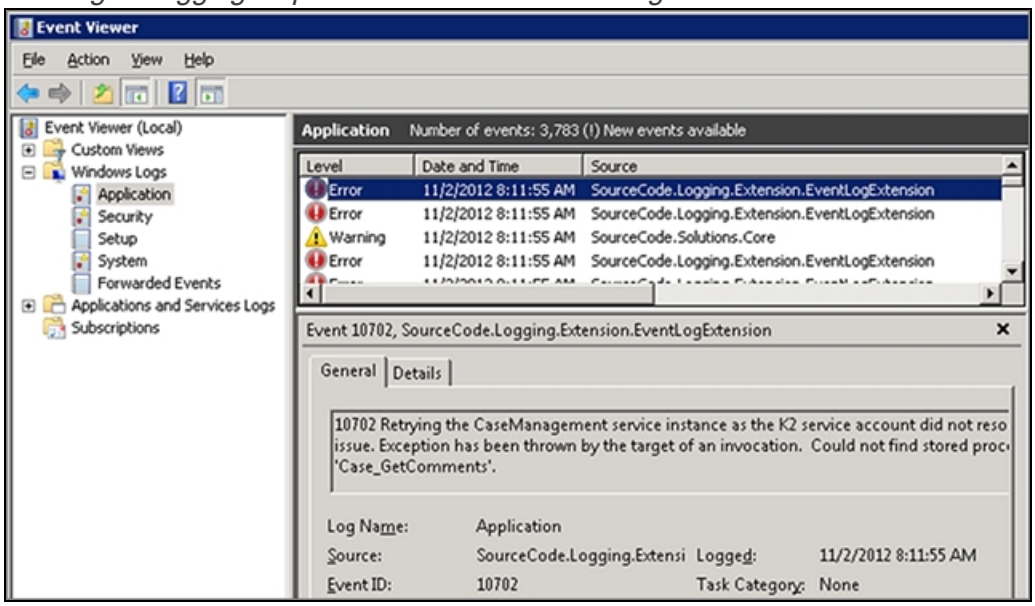


Setting permissions with K2 Workspace



In addition to the administration interfaces, K2 also outputs logging information to specified targets such as text files or the window event viewer. Administrators can use logging output to check the health of the K2 environment, discover and troubleshoot problems or otherwise set up some type of monitoring to be notified of issues on the K2 environment.

Viewing K2 logging output in the Windows Event Log

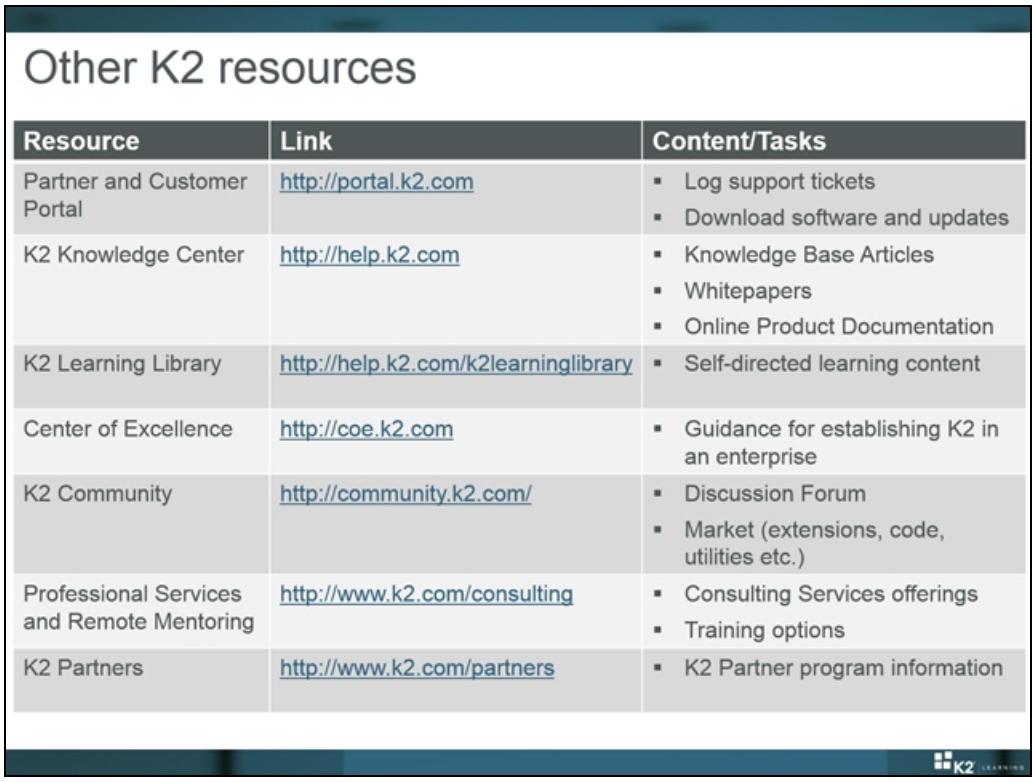


Summary

- Administrators use K2 Management to administer a K2 environment.
- Typical workflow instance admin tasks include suspending and resuming workflow instances or deleting workflow instances and repairing workflows that have failed.
- Typical task admin tasks include redirecting and releasing tasks and setting up out-of-office rules on behalf of users.
- Typical environment admin tasks include setting up security and permissions for workflows and the K2 environment as a whole, and configuring the environment (such as setting up service instances, licensing, and certain operational parameters).

- K2 Admins can configure the logging output targets (e.g. text files, windows event log, custom) and review that output to monitor and troubleshoot K2.

Other K2 resources



Video

After completing your initial training with K2, several services and resources are available to help you achieve the most with your K2 implementation.

Partner and Customer Portal

K2 Partners and Customers have access to a dedicated self-service portal (http://portal.k2.com). Access to the Customer and Partner portal is restricted to registered K2 customers and partners, and you will require a user account to access the resources in the portal. After logging into the portal, you may download product updates, submit and search your organization's support tickets, provide feedback on a K2 product, manage your K2 licenses and manage the portal user accounts for your organization.

K2 Knowledge Center

Our help site (http://help.k2.com) provides access to Knowledge Base (KB) articles, technical whitepapers, training videos and training courses, an online version of the K2 product documentation as well as .pdf versions of the K2 documentation.

K2 Learning Library

K2 Learning is a division of K2 that creates training material to empower users of the K2 platform. (This training module is an example of the content maintained by K2 Learning.) You can access K2 Learning materials through the K2 Learning Library in the Knowledge Center (http://help.k2.com/k2learninglibrary) or your K2 representative. Instructor-led K2 training courses are offered based on the material created by K2 Learning, and certain materials are available for self-study.

Center of Excellence

The K2 Center of Excellence (http://coe.k2.com) is a repository of standards, practices, whitepapers and guidelines that will help enterprises establish K2 as a solid workflow and business process management platform. The content in the Center of Excellence is based on many years of practical experience from K2's field organization, implementing large, complex projects in enterprise customers.

K2 community

The K2 Community (http://community.k2.com) is an open, community-managed site with content such as forums and blogs. We recommend that you sign up and join the K2 community, since there is a lot of useful content on the K2 Underground to help you in your K2 projects. The K2 forums are useful for research and to ask the community for help or suggestions, assuming you do not wish to make use of K2 support or Technical Advisory Services.

The K2 community site also features downloadable code samples and prebuilt components offered by the K2 community. These components are not supported through the normal K2 support channels, but there are a lot of useful components that could help speed the delivery of K2 solutions in your projects.

Professional Services and Remote Mentoring

K2 can provide on-site or remote technical consulting services to K2 customers and partners. K2 Professional Services consultants are very well versed in the suite of K2 products, and have extensive industry and technical experience in many enterprise-level technologies and products.

K2 also offers a Remote Mentoring service that provides affordable and requirement-specific guidance and help. This service provides an alternative to a full-fledged consulting engagement, and is typically provided over short timeframes to address a particular requirement and issue. Contact your K2 representative or nearest K2 office if you wish to make use of K2's Professional Services or Remote Mentoring, or see http://www.k2.com/consulting.

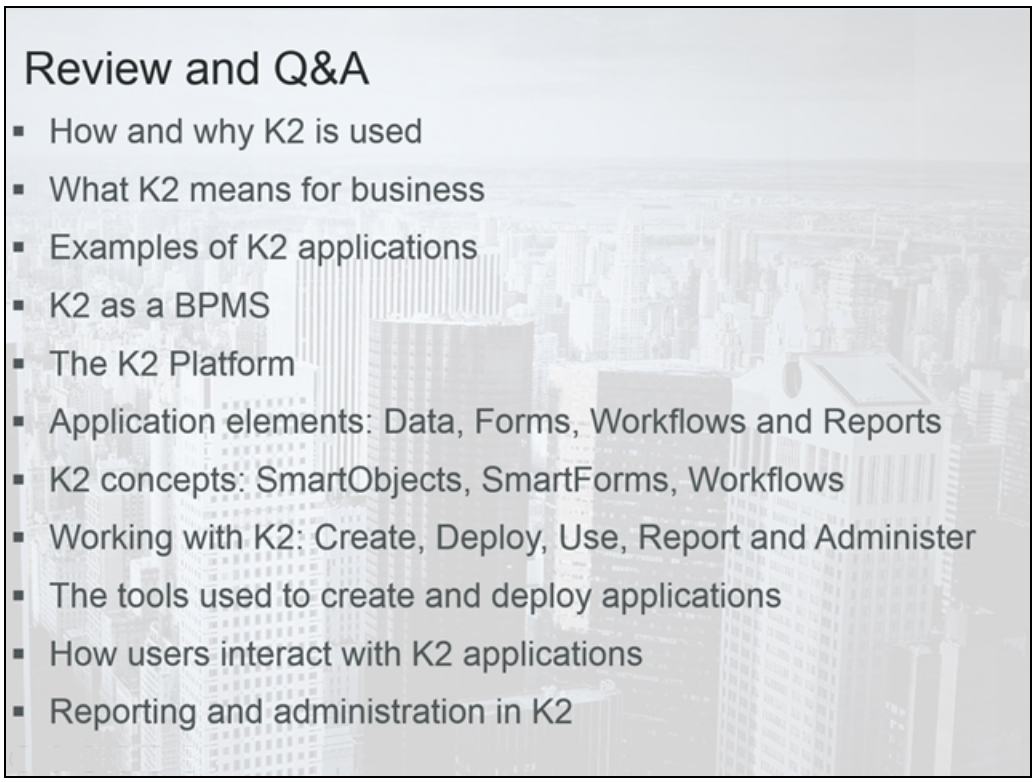
Partners

K2 has partnership agreements with external service organizations to resell and deliver K2 products. K2 partners have dedicated account management contacts and have direct access to K2 resources for support, if required. Because of the breadth, scope and potentially industry-specific implementation of K2 solutions, K2's preferred delivery channel for K2 solutions is through our Service Provider Partners. Partners can access dedicated partner resources at http://www.k2.com/partners.

Summary

- There are many resources that people can use to get help or more information on K2
- Some of these resources require a K2 portal login account, others do not
- The community site is provided by K2 but not supported by K2
    - Any sample code, utilities, samples, extensions, etc., downloaded from the community site is not maintained or supported by K2

Review and Q&A: Introduction to K2



Video

This topic is just a summary of the information covered in this module: Introduction to K2. If you are attending a training session led by a K2 instructor, this is your opportunity to ask questions or, if time permits, discuss your own plans for using K2 in your environment.

In summary, here is what we covered in this training module:

- What K2 means to enterprises
- How K2 is used and the benefits of using K2
- K2 as a Business Process Management System
- The components of the K2 platform
- K2 applications are made up of Data, Forms, Workflows and Reports
- Foundational K2 concepts like SmartObjects, SmartForms and Workflows
- What tools and interfaces are used to Build, Deploy, Use, Report on and Administer K2
- How users interact with K2 applications

Knowledge-check and review questions

To check your own knowledge and comprehension, consider the questions and answers below. Some are specific to K2, others may require you to think how K2 applies to your environment.

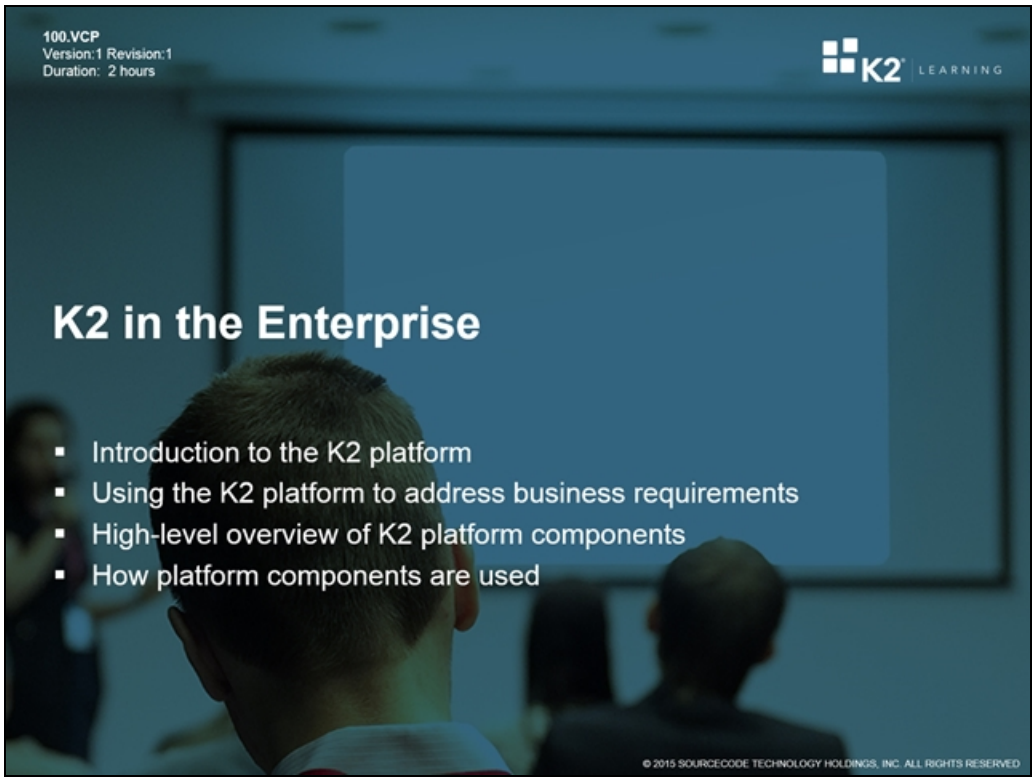
Q: How can users interact with K2 applications?
Reveal answer**A:** SmartForms and other user interfaces, K2 SmartActions (reply by email), use mobile applications to complete tasks.

Q: What tools can be used to build workflows?
Reveal answer**A:** K2 for SharePoint , K2 Designer, K2 Studio, K2 for Visual Studio

Q: Based on your own role in your organization, which tools would you most likely be using? Workspace? K2 Studio? The web-based designers? And why?
A: (discussion question, no right/wrong answer)

Q: Based on your organization, do you have specific projects that you intend to use K2 for? Or can you identify areas and solutions where K2 might be a good platform to implement a solution?
A: (discussion question, no right/wrong answer)

K2 in the Enterprise

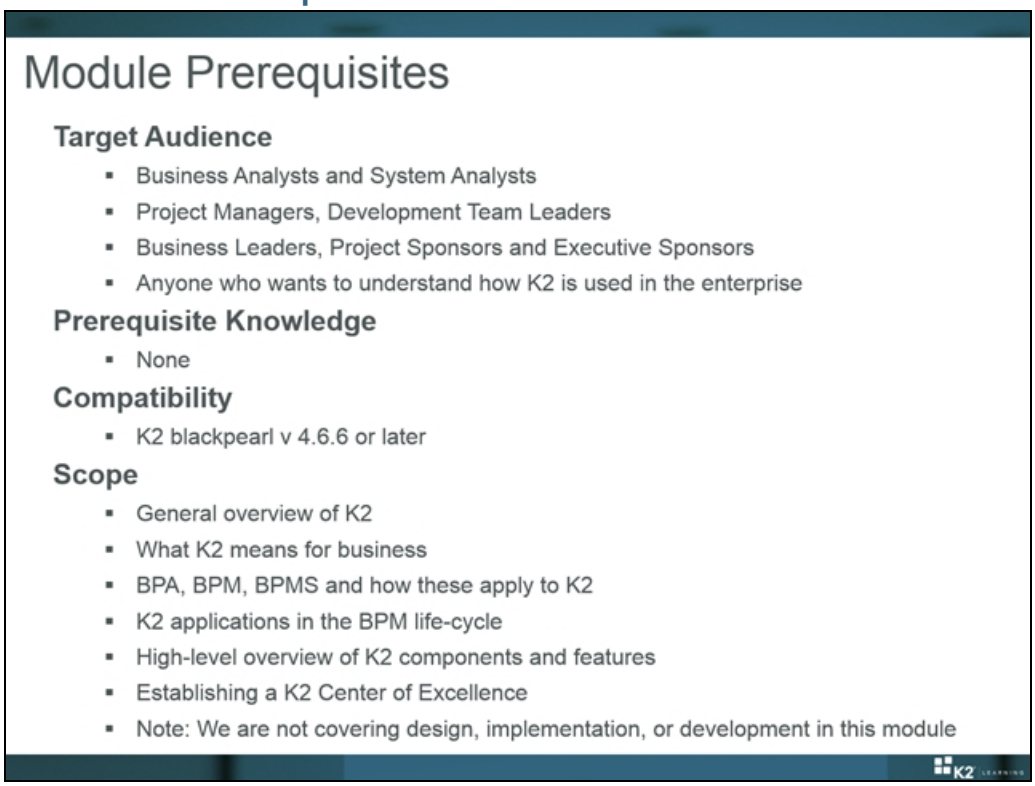


Welcome to K2 in the Enterprise. This module builds on concepts introduced superficially in the module 100.ABZ – Introduction to K2 and should take about 2 hours to complete. Roughly 50% of time will be spent on lecture and 50% on activities and exercises.

By the end of this module, you should be able to:

- Explain how the K2 platform addresses business requirements
- Describe K2 platform components in general
- Define BPA, BPM, and BPMS
- Explain why organizations use BPM and BPMS
- Explain why organizations use K2 for BPM and BPMS
- Explain what SmartObjects are and what they do
- Identify the five phases of BPM maturity development in organizations
- Identify the six stages in the BPM life-cycle

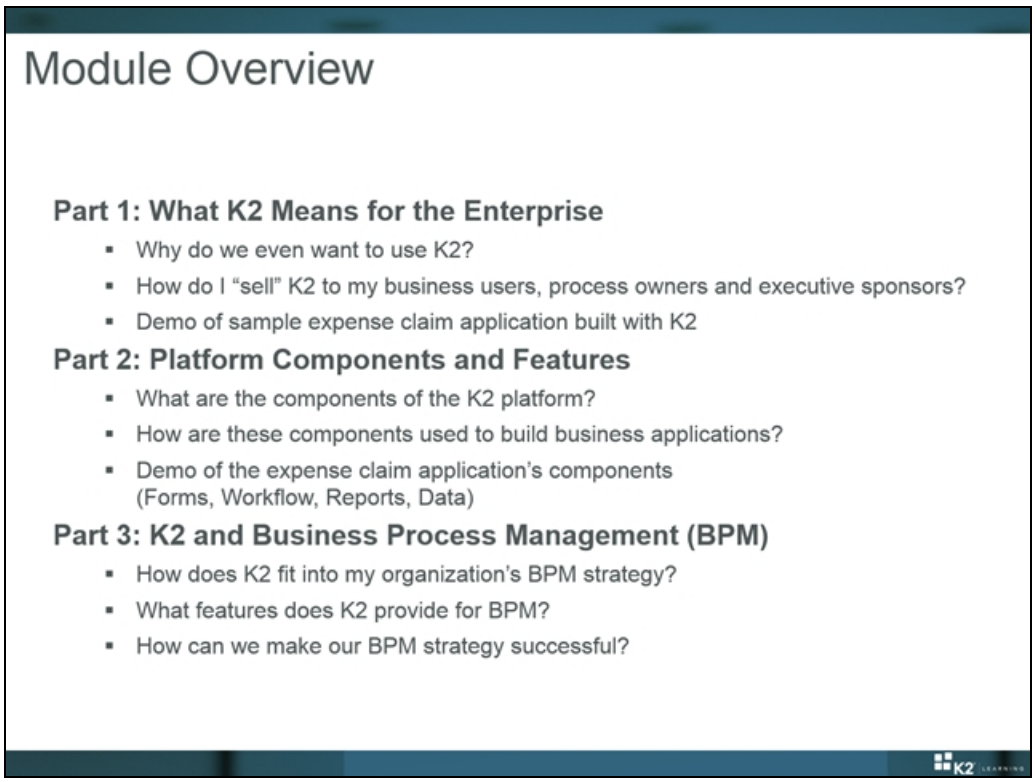
Module Prerequisites



If you are a business analyst, system analyst, project manager, development team leader, business leader, or other sponsor, this introductory module explains how K2 fits into the enterprise. We are not considering design or implementation yet.

Note
If you are taking this course in a webinar or other online format, your instructor may ask for your participation in the small group activities on slides 5 and 10.

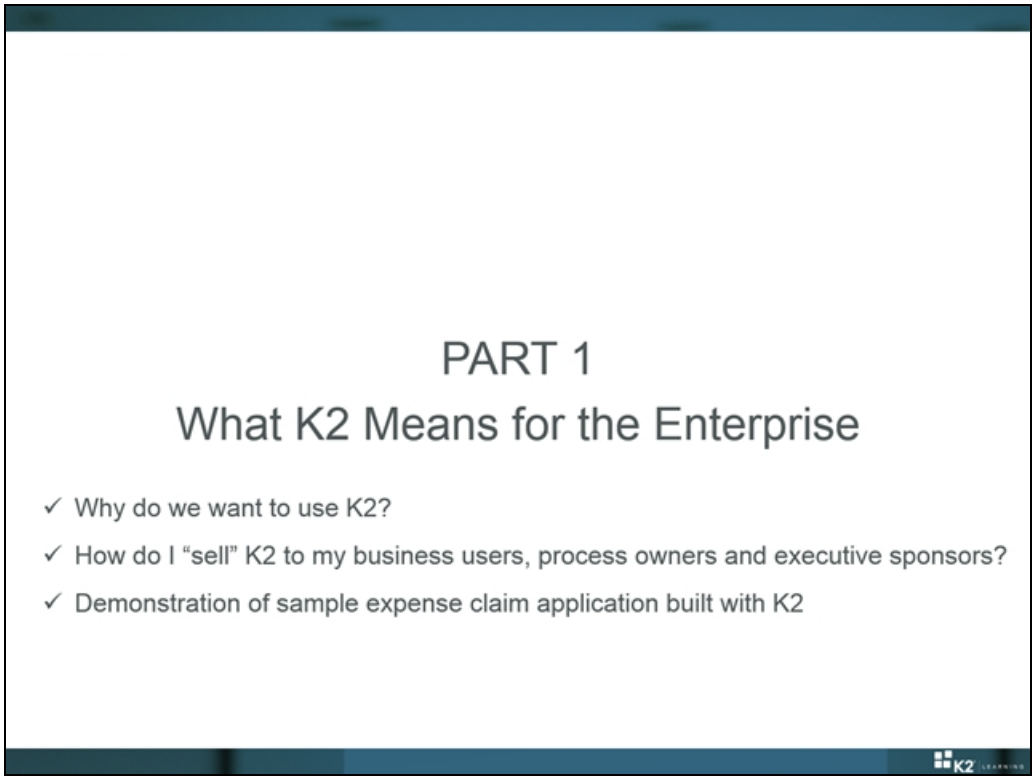
Module Overview



This module is arranged in three parts. Take a moment to consider these parts as an overview of the module:

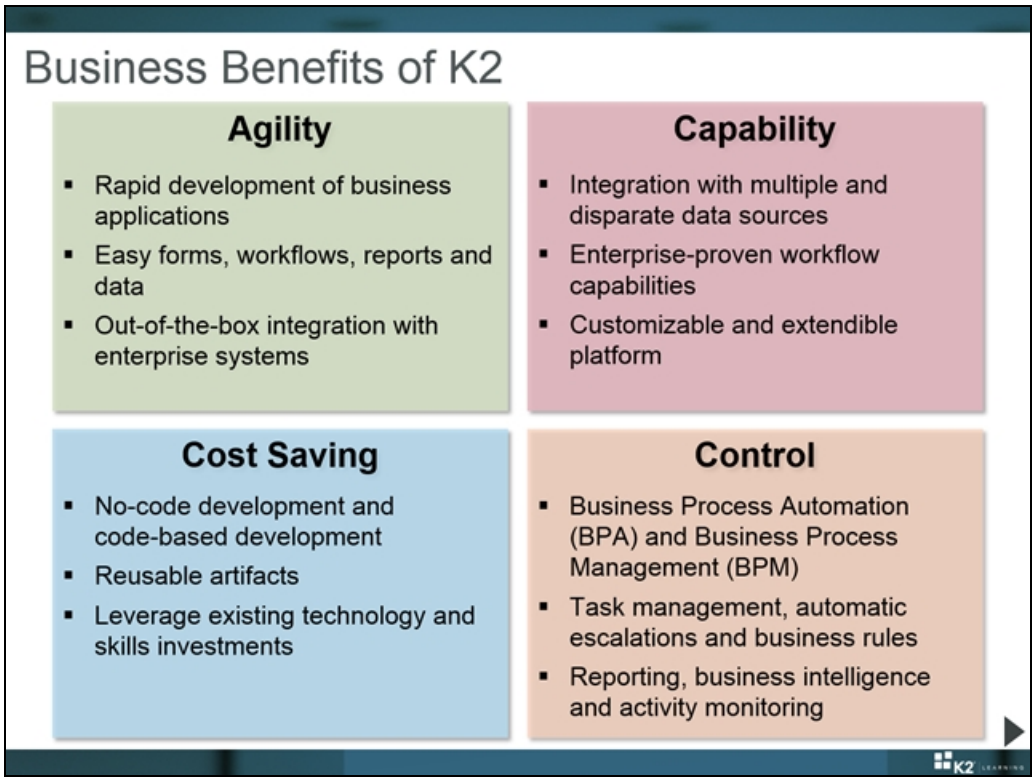
- Part 1 focuses on what K2 means for the enterprise in terms of:
  - Business benefits and examples of business applications
  - K2 scope
  - BP terminology
  - BPM and BPMS
  - How K2 helps organizations with BPM and BPMS
- Part 2 dives into:
  - The K2 platform
  - SmartObjects
- Part 3 explores:
  - How organizations mature in the use of BPM
  - Building a Center of Excellence
  - The stages of the project life cycle

PART 1: What K2 Means for the Enterprise



In Part 1 we will look at the business benefits of K2. Next we will practice selling the business benefits of K2 to business users, process owners, and executive sponsors. At the end of Part 1, we will see a demonstration of a sample expense claim application.

Business Benefits of K2



Perhaps at this point you're wondering, “Why should I care about K2?” or “What’s in it for me/us/our enterprise?” Those are good questions and perhaps your colleagues are wondering the same things. The activity below can help prepare you to “sell” K2 to my business users, process owners, and executive sponsors.

Small Group Activity: 3-5 minutes x 2

If you are in a **live class** environment, your instructor may direct you to:

- Divide into groups of four.
- The instructor will give each group member an index card bearing the name of one of the benefit groups shown here.
- Pair up with a member of your team and “sell” or deliver a 30-second sales pitch for one of the benefits from your card to your partner.
- If you make mistakes, pause and restart, or ask your partner for his or her ideas on presentation.
- After a few minutes delivering your "pitch," switch roles and let your team member deliver his or her pitch.

If you are in a **webinar** environment, your instructor may direct you to:

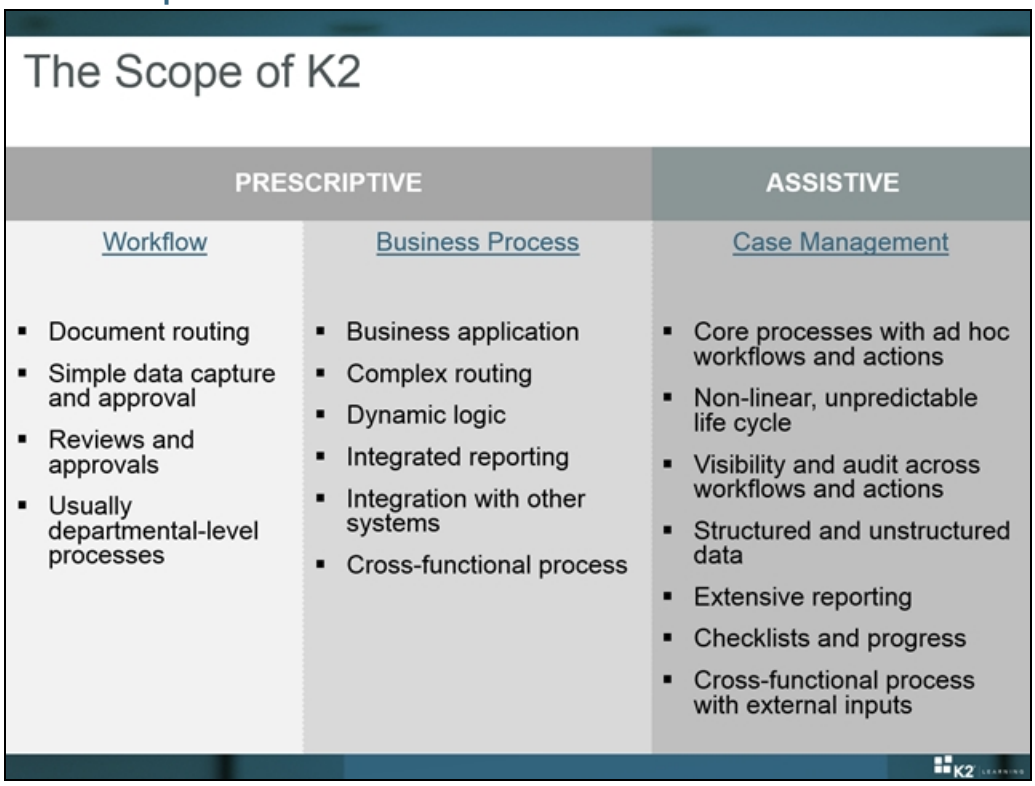
- Study the four benefit groups or think of benefits that might apply to one of these groups from your organization.
- Volunteer to share your benefit with the group for discussion.
- Discuss as a group ways that your benefit can be "pitched" to stakeholders.

If you are in a **self study** environment, you may:

- Study the four benefit groups and choose or think of a benefit that might apply to your organization.
- Write a 30-second sales "pitch" that sells your chosen benefit to a prospective stakeholder.
- Ask a coworker to listen to your 30-second pitch after you have refined it for presentation.
- Encourage your coworker to critique your pitch.

Note
Avoid confusing benefits with features. Contrast the following:
Feature - the Blastar 5000 has a keyless entry system
Benefit - no more fumbling for your keys in the dark

The Scope of K2



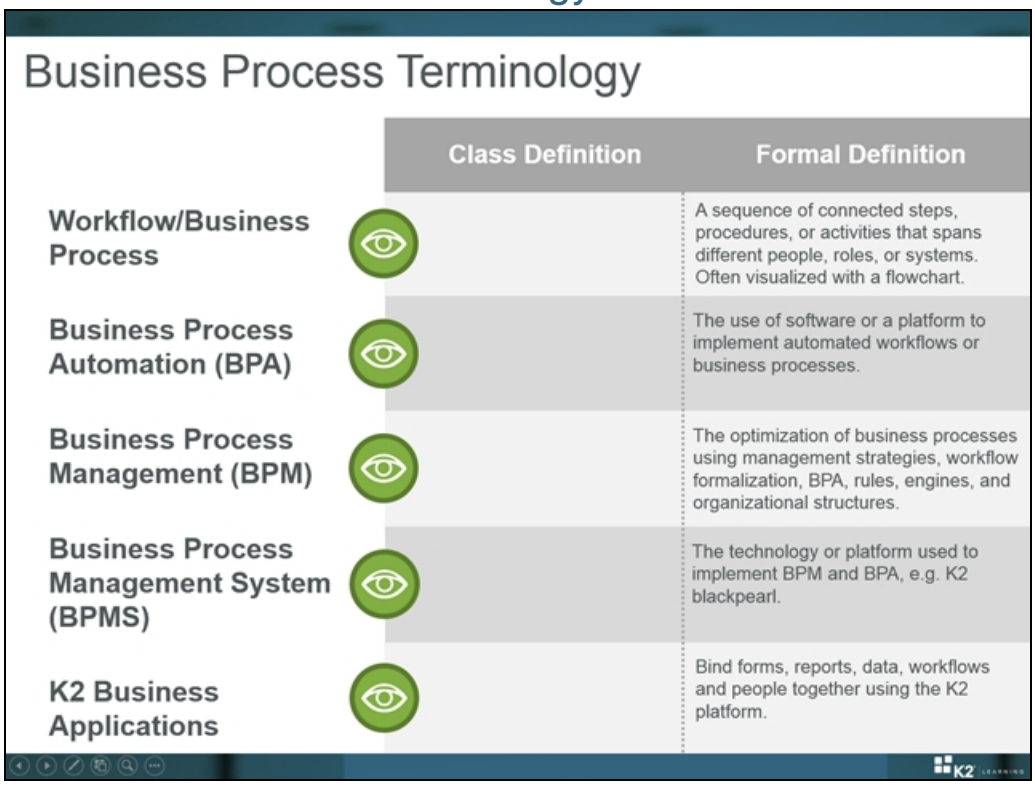
Take a moment to consider the heading terms "**prescriptive**" and "**assistive**". These terms serve as effective descriptions of the spectrum of workflow types covered by K2.

The scope of K2 spans from **departmental workflows** to organizational **business processes** with integration into various systems and cross-functional to **case management** where processes are unpredictable and may include resources outside of the organization.

Your class members may discuss the following:

- Who can give me an example of **document routing**? (Choose one or more of the items in the list and ask learners to cite examples.)
- Who can give me an example of a **business application**? (Choose one or more of the items in the list and ask learners to cite examples.)
- Who can give me an example of **core processes with ad hoc workflows and actions**? (Choose one or more of the items in the list and ask learners to cite examples.)
- Mention that even though it is not officially supported, there is a Case Management add-on available for K2 to help organizations build **Case Management** solutions.
- Even if you don't have the Case Management Framework (CMF) installed, you can still build your own Case Management solutions.
- The point is that K2's capabilities extend across all these different "levels" of business process complexity.

Business Process Terminology



This slide helps establish a common base of business process terminology. If you are in a **live or webinar** class environment, your instructor may ask learners for their definitions of each term.

If you are in a **self-study** environment, consider the following:

Workflow/Business Process - a sequence of connected steps, procedures, or activities that spans people, roles, or systems to accomplish business purposes or goals. They are often visualized with a flowchart.

Business Process Automation - the use of software or a platform to implement automated workflows or business processes.

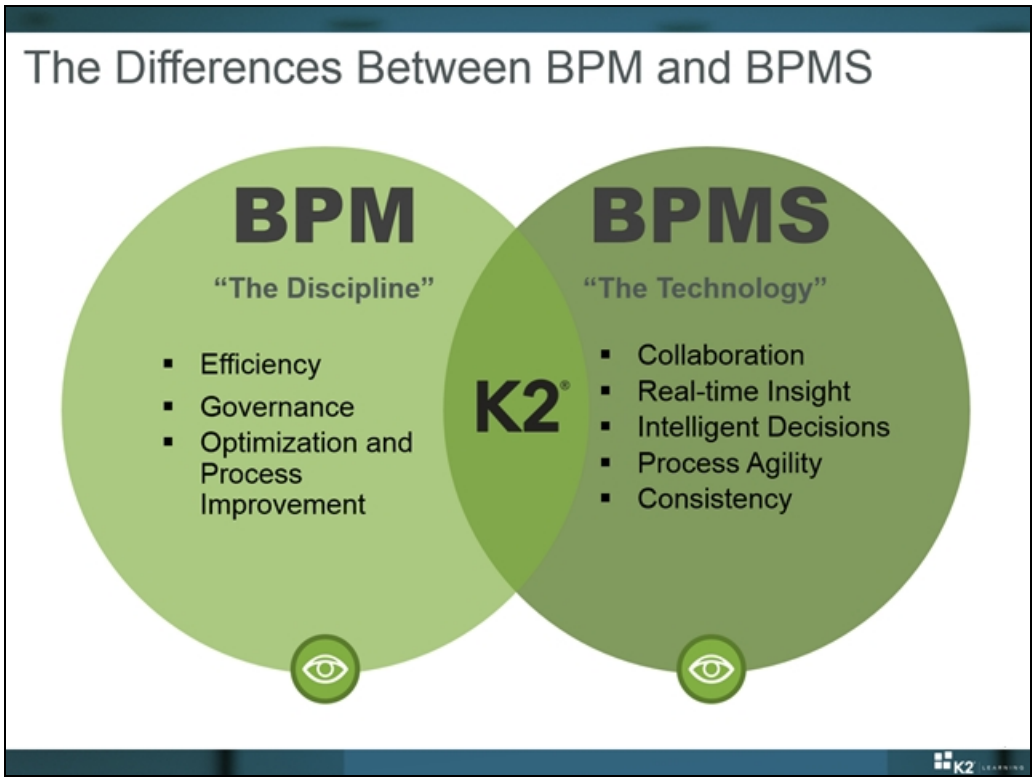
Business Process Management - the optimization of business processes using management strategies, workflow formalization, BPA, rules, engines, and organizational structures.

Business Process Management System - the technology or platform used to implement BPM and BPA such as K2 blackpearl.

K2 Business Applications - binding forms, reports, data, workflows, and people together using the K2 platform.

Normally, in a live class or webinar students have a chance to discuss and compare their individual work situations and whether these definitions apply to those environments. What do you think? On a scale of 1 to 5, 5 being "very closely" and 1 being "not at all," how well do you think these definitions apply to your own business setting?

The Differences Between BPM and BPMS



Let's discuss the differences between "BPM - The Discipline" and "BPMS - The Technology".

Business Process Management is an approach to business optimization, whose objectives are to empower people, improve productivity, and increase efficiency through optimization, process improvement, and governance. It is much more than just automating business processes with a BPA platform.

BPMS is a technology that ultimately aims to enable real-time collaboration and insight to:

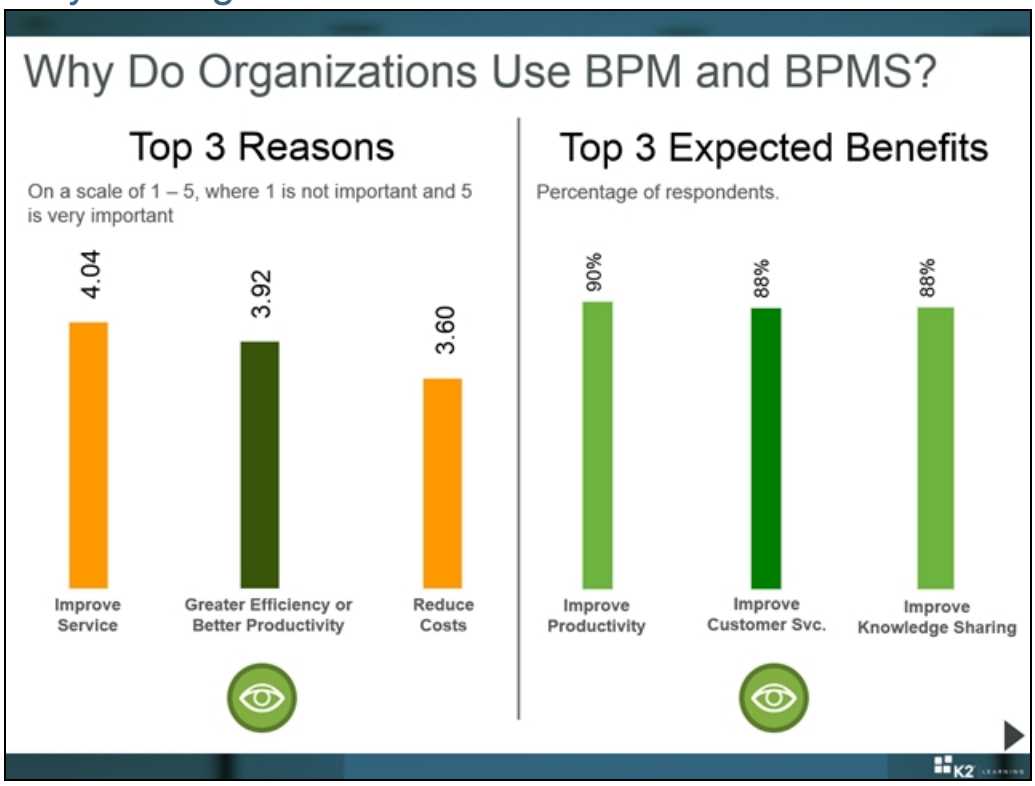
- Drive informed decision and outcomes
- Support process agility and consistency

The point is that K2 supports both. It is a BPMS platform that enables BPM. However, K2 doesn't do everything: BPM requires additional input and effort from the organization. That is not to say that BPM is a prerequisite for K2 solutions.

Simple K2 solutions can run fine without a convoluted or extensive BPM discipline. But for large-scale enterprise projects, you should take BPM into consideration to get the best results. BPM by itself a huge area and we can only touch on it in this training content.

There are many more resources online and books that cover BPM at a deeper level, but for the purposes of this training we need to clarify how K2 supports the principles of BPM.

Why Do Organizations Use BPM and BPMS?



Let's look at some results of a survey performed by PMP research that asked organizations to rate:

- The reasons why they want to use BPM and BPMS
- The expected benefits of implementing BPM and BPMS.

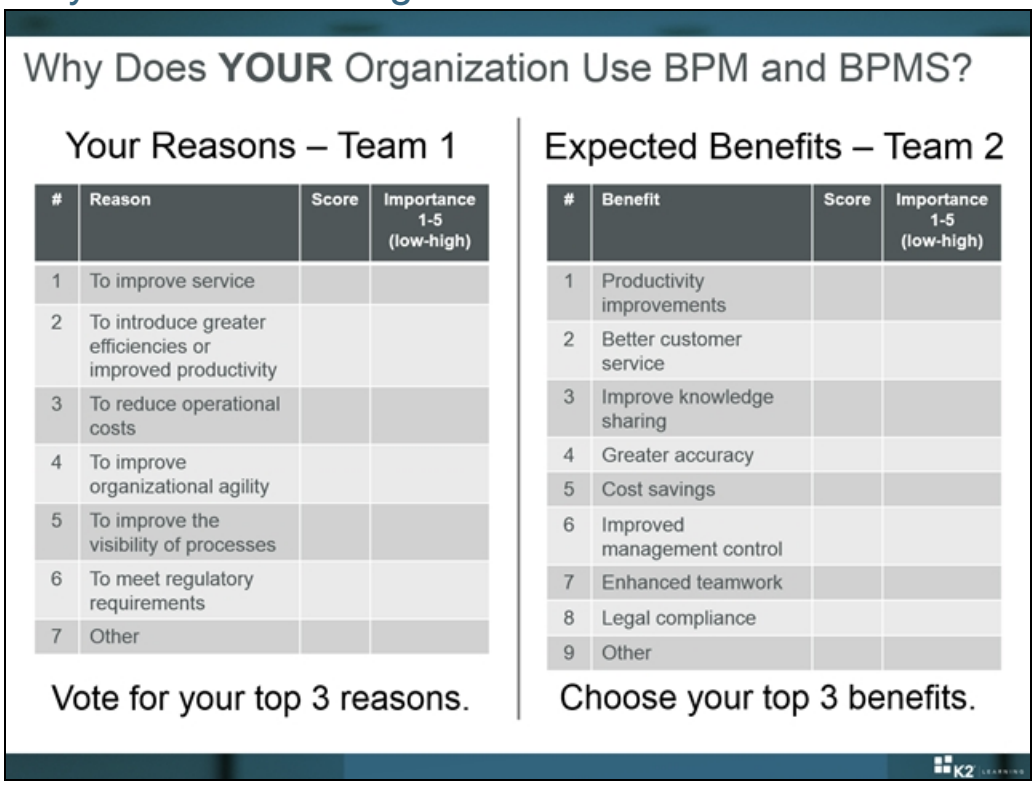
The Top 3 Reasons>

- To improve service
- To achieve greater efficiency or better productivity
- To reduce costs

The Top 3 Expected Benefits>

- To improve productivity
- To improve customer service
- To improve knowledge sharing

Why Does YOUR Organization Use BPM and BPMS?



In a **live class** environment, this is a small group activity in which you divide the class into two smaller teams. Each team then discusses either topic 1 (Why their organization uses BPM and BPMS) or topic 2 (What benefits they expect from using BPM and BPMS). After brief discussion, each member votes for his or her top three reasons or benefits. The class reforms and the results are recorded (and discussed as time allows).

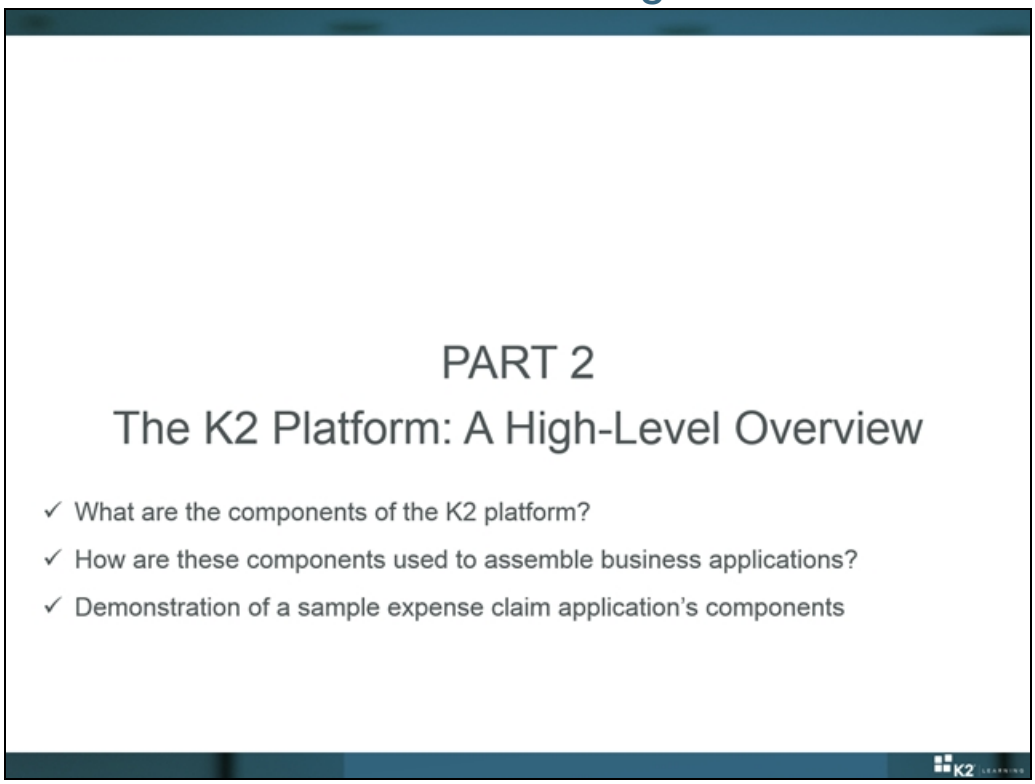
In a **webinar** environment, the instructor may poll participants by means of the vote feature in Saba, then record the results.

Let's look at some of **your** organization's reasons for using and benefits from BPM and BPMS.

Small Group Activity: 5-7 minutes

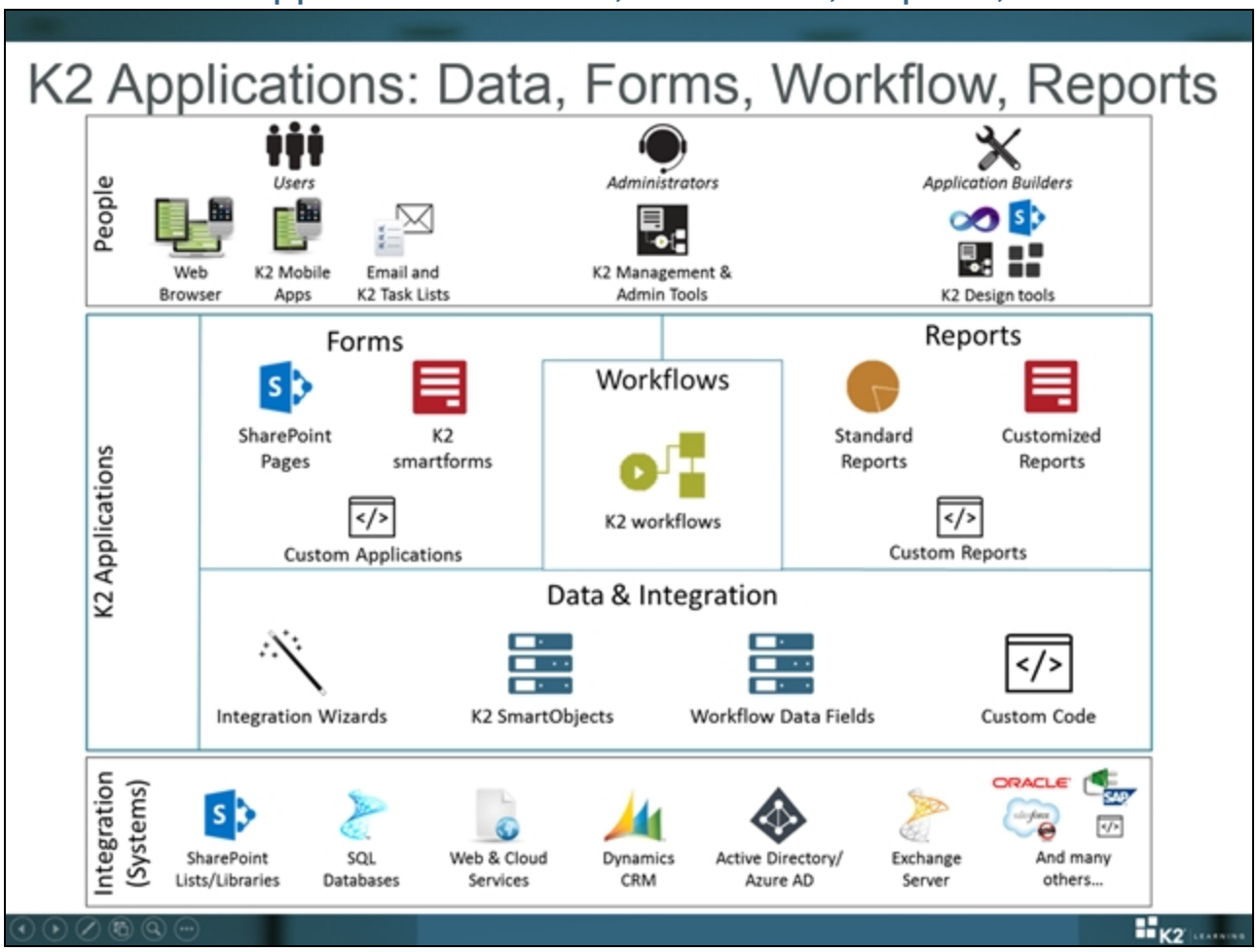
- In a **live** environment, divide the class participants into two groups (Team 1 – Reasons and Team 2 – Benefits).
- In a **webinar** environment, use your vote button to choose your top three reasons and top expected benefits.
- Give each group sticky note pads. **live**
- Ask each team member to take a sticky note and write their organization's top 3 reasons (team 1) or top 3 expected benefits (team 2) on it. **live**
- Team members can also assign a priority (1-3) for each item they vote for. **live**
- Give teams 4-5 minutes to discuss the various reasons and benefits. **live**
- Conclude the activity by scoring each team's responses. **both live and webinar**

PART 2: The K2 Platform: A High-Level Overview



In Part 2 we will look at the components of the K2 platform. We consider how these components are used to assemble business applications. At the end of Part 2, we will see a demonstration of a sample expense claim's components.

K2 Business Applications: Forms, Workflows, Reports, and Data

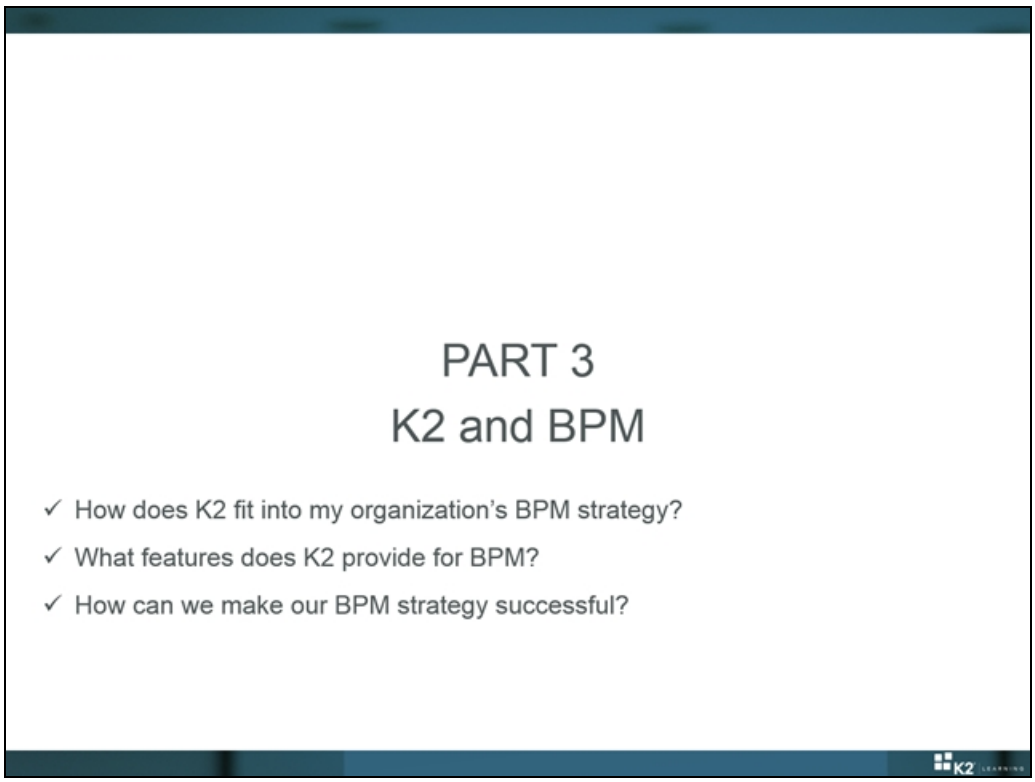


K2 business applications consist of Data, Forms, Workflows, and Reports shown here in a layered application architecture.

Forms provide user interfaces for interacting with a K2 application. You could use K2 smartforms, Microsoft InfoPath forms, custom web apps written in something like ASP.NET, or SharePoint user interfaces. It all depends on your requirements. K2 has a selection of standard Reports that are primarily focused on workflow reporting. Workflows are implemented as K2 workflows, using any of the available K2 design tools. The Data component of a K2 application is normally implemented as K2 SmartObjects. These SmartObjects can expose a wide range of common enterprise systems such as SQL, Active Directory, Exchange, SharePoint, and others as data providers. You're not limited to the standard connectors, though. You can create custom connectors (called "Service Brokers"), which can expose other technologies as K2 SmartObjects as well. All these components work together to create rich, powerful and flexible business applications.

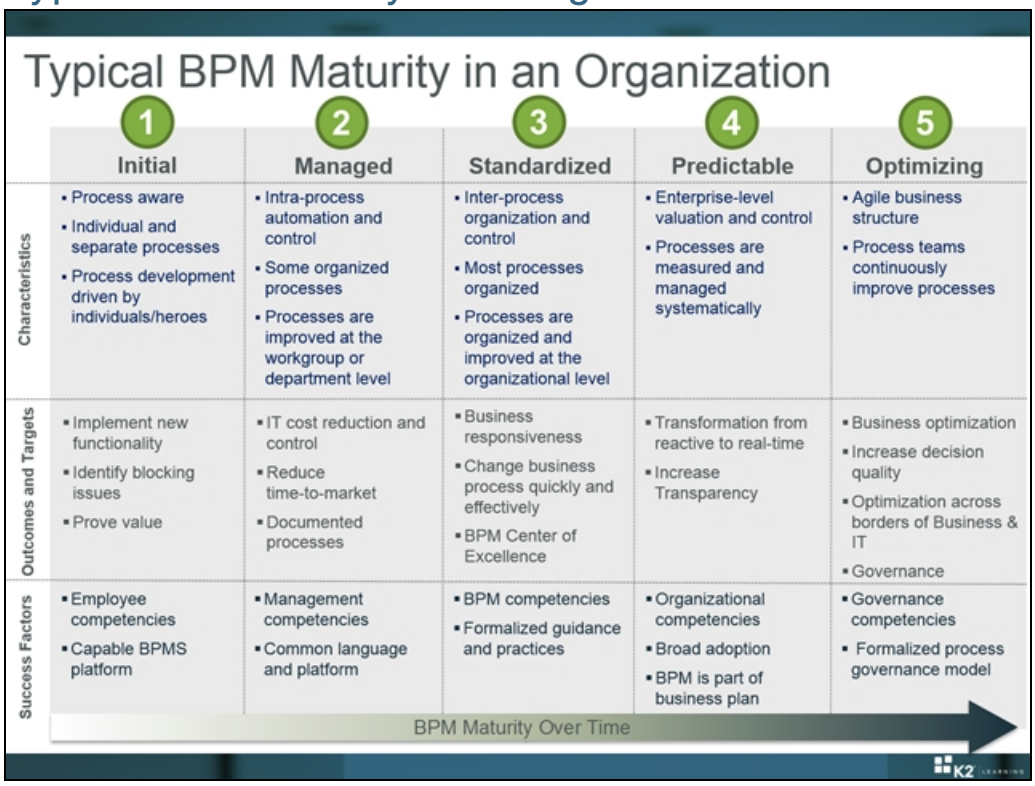
K2 provides standard integration with various enterprise technologies .

PART 3: K2 and BPM



Up to this point, you've learned what K2 means for business and you've seen a sample solution built on K2. In Part 3 we will look at how K2 supports BPM. This section of the module is especially important for organizations that are just starting out with BPM, or already have BPM initiatives under way.

Typical BPM Maturity in an Organization



BPM is not a do-it-once, 5-minute fix. Instead, organizations typically mature in BPM over time, and we can divide this up into 5 main "levels" of maturity. It's not always necessary to get from Level 1 to Level 5 in all situations. Many organizations remain at level 2 or 3 for some time, and perhaps their turnover, resourcing or requirements means that they do not need to move up to level 5. That is not a bad thing. After all, the **needs should drive the implementation**, not the other way around.

What is important to note is that BPM is a journey, and successful BPM implementation usually starts small and works it way up the levels of maturity over time. Trying to go straight into a Level 4 or 5 of BPM maturity from nothing is usually not successful, you need to build up skills, expertise, and governance over time.

It is not necessary to go though each of the 5 levels in detail, just explain that the levels basically boil down to this:

1) **Initial** – small projects, prove the platform, driven by individuals.

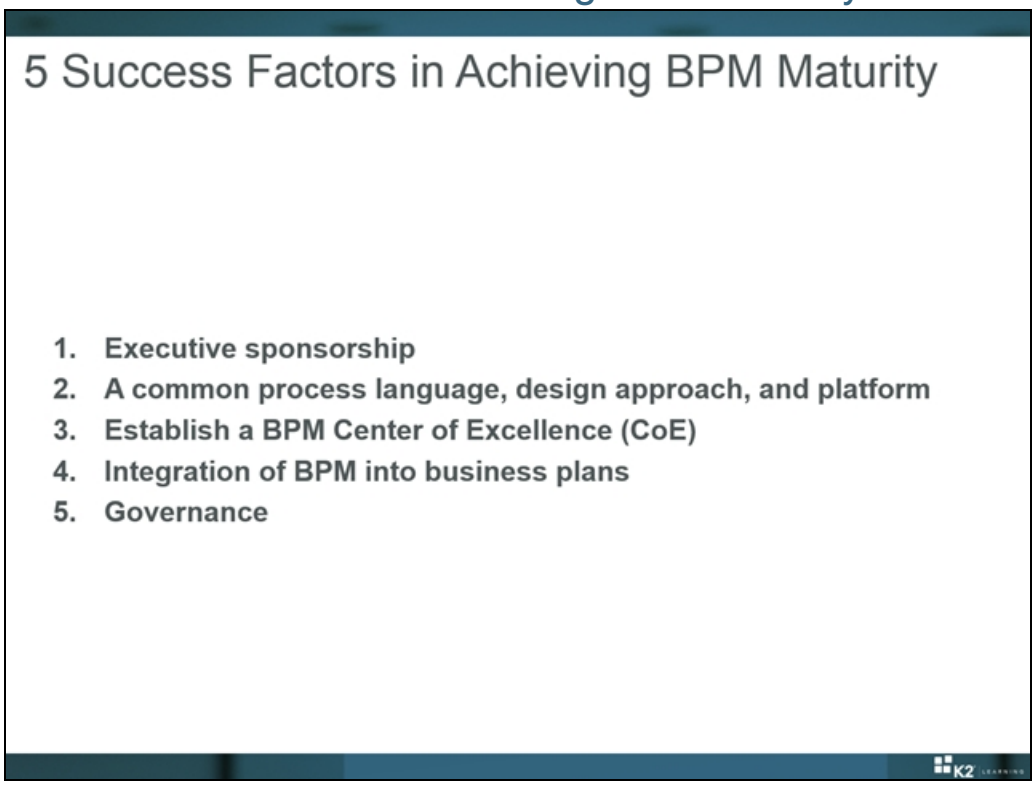
2) **Managed** – more structured processes, inter-process and departmental processes. Some new competencies in managing BPM projects.

3) **Standardized** – the organization has standard approaches for BPM projects and control such as conventions and design standards. Inter-departmental processes start to appear. This is a good time to establish your BPM center of excellence.

4) **Predictable** – the BPM platform is stabilized, the organization has standard procedures and practices for developing and maintaining BPM projects, the CoE is actively involved with BPM projects, and BPM becomes part of the organizations business plan for the future. New projects are more predictable in length and complexity, and a lot of the underlying infrastructure should already be in place for these new projects.

5) **Optimizing** – BPM is entrenched in the organization, and the guidelines, standard and governance is in place. At this point, the focus is on optimizing: old BPM projects can be optimized, the overall flow of BPM development projects is optimized, business rules are tweaked to respond to changing requirements, and generally BPM becomes an integral part of the organizations operation and strategy.

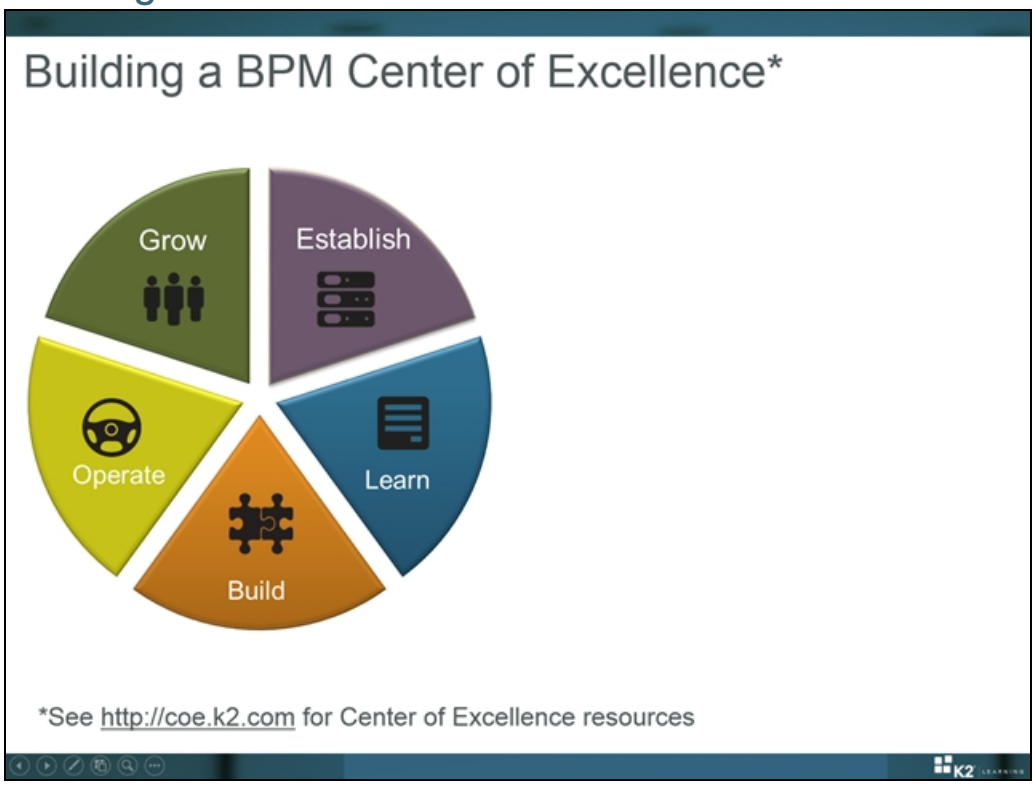
5 Success Factors in Achieving BPM Maturity



The 5 most important factors in achieving BPM maturity are:

1. **Executive sponsorship**. A lack of vision and executive buy-in and sponsorship can kill a BPM initiative easily.
2. **Common process language**, design approach and platform. Using all these common items will make it much easier to implement BPM projects and ensure consistency and compliance with know best practices. This is very important!
3. **Establishing a BPM Center of Excellence**. This is the go-to team and set of resources to ensure that BPM projects are consistent, controlled, and implemented correctly according to the organization's implementation of BPM.
4. **Integration of BPM into business plans**. This is important because otherwise, BPM becomes just another methodology or esoteric management theory. If BPM is integrated in the business plan, it becomes part of the organization's culture and success.
5. **Governance** is important to ensure that the BPM standards are maintained and controlled. Governance is about control, guiding, and reigning-in. It is easy to get carried away and easy for BPM projects to start multiplying beyond control. Good governance is crucial to ensure that the recommended path is always followed. Governance includes setting direction and defining the BPM strategy in alignment with the business strategy, implementing controlling resources like a governance board, enabling the CoE with resources to execute on the strategy, defining the BPM priorities for the organization and a standard mechanism for BPM program plans.

Building a BPM Center of Excellence

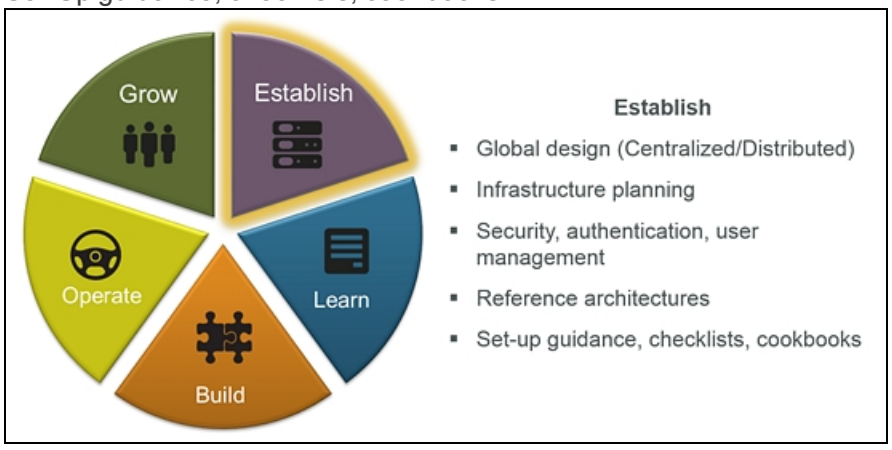


Building a BPM Center of Excellence can be a daunting task, but it becomes easier if you start to break down the process into multiple stages. This slide illustrates a recommended approach for establishing a CoE, all the way from establishing your CoE to growing the CoE to maturity.

Your local K2 office or K2 technical representative may be able to assist you in developing your CoE. You should also refer to the Center of Excellence website (http://coe.k2.com) which contains many resources that will help you and guide you in establishing each of the CoE principles in your organization.

Establish

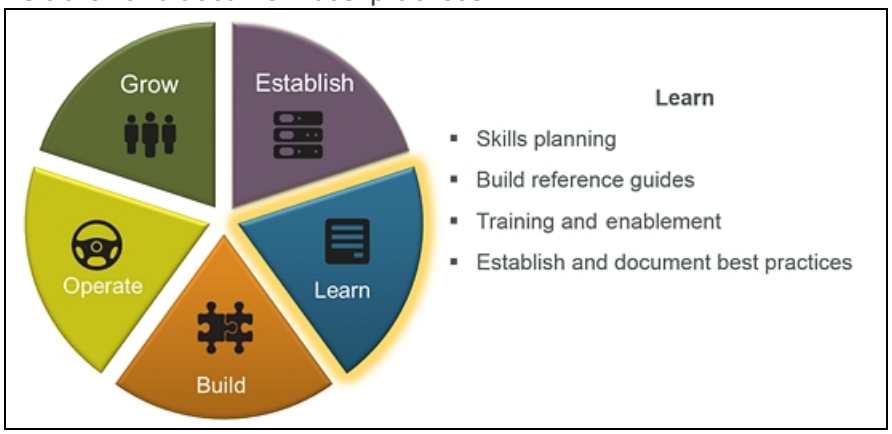
- Global Design (Centralized/Distributed)
- Infrastructure planning
- Security, authentication, user management
- Reference architectures
- Set-Up guidance, checklists, cookbooks



Learn

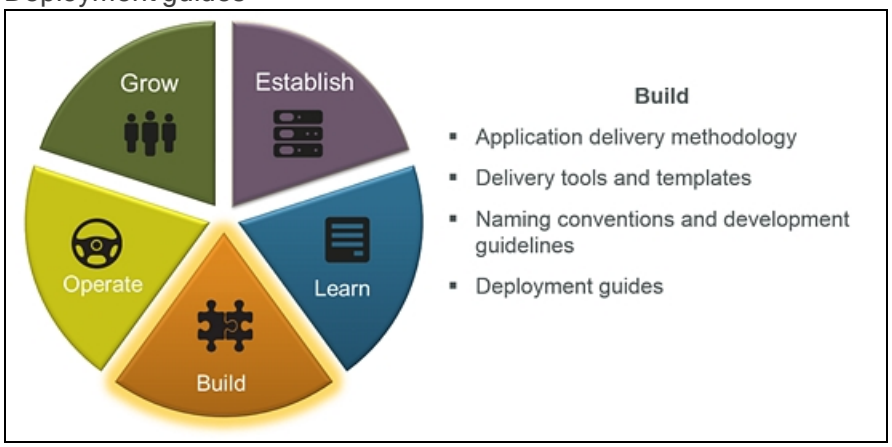
- Skills planning
- Build Reference guides

- Training and Enablement
- Establish and document best practices



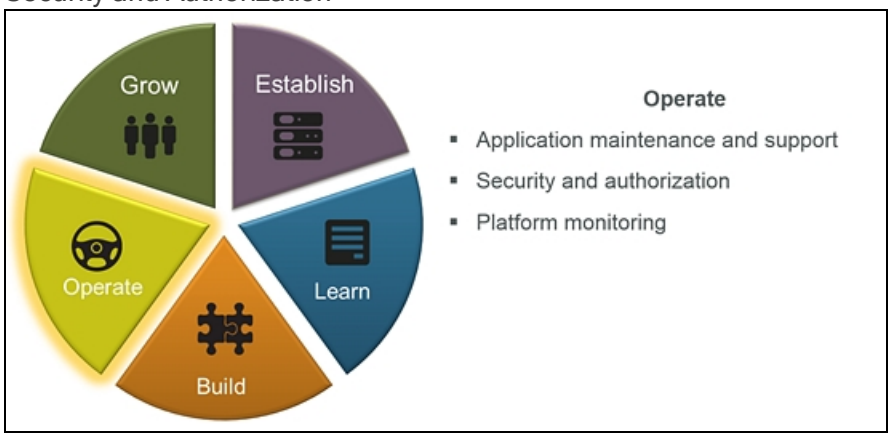
Build

- Application delivery methodology
- Delivery tools and templates
- Naming conventions and development guidelines
- Deployment guides



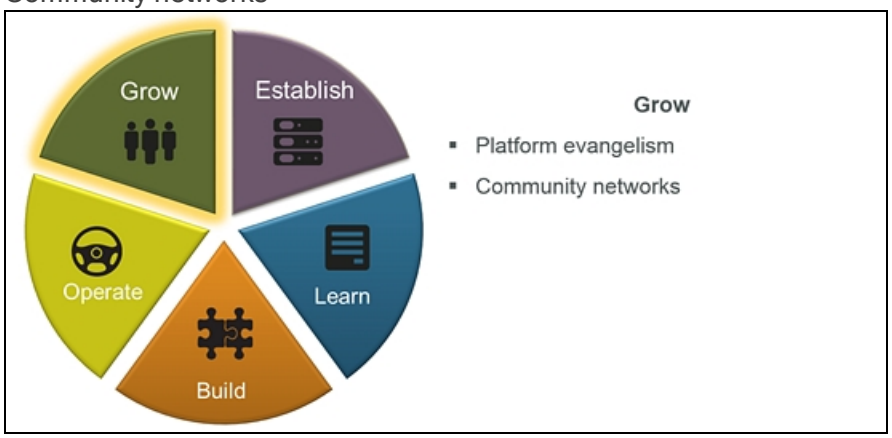
Operate

- Platform monitoring
- Application maintenance and support
- Security and Authorization

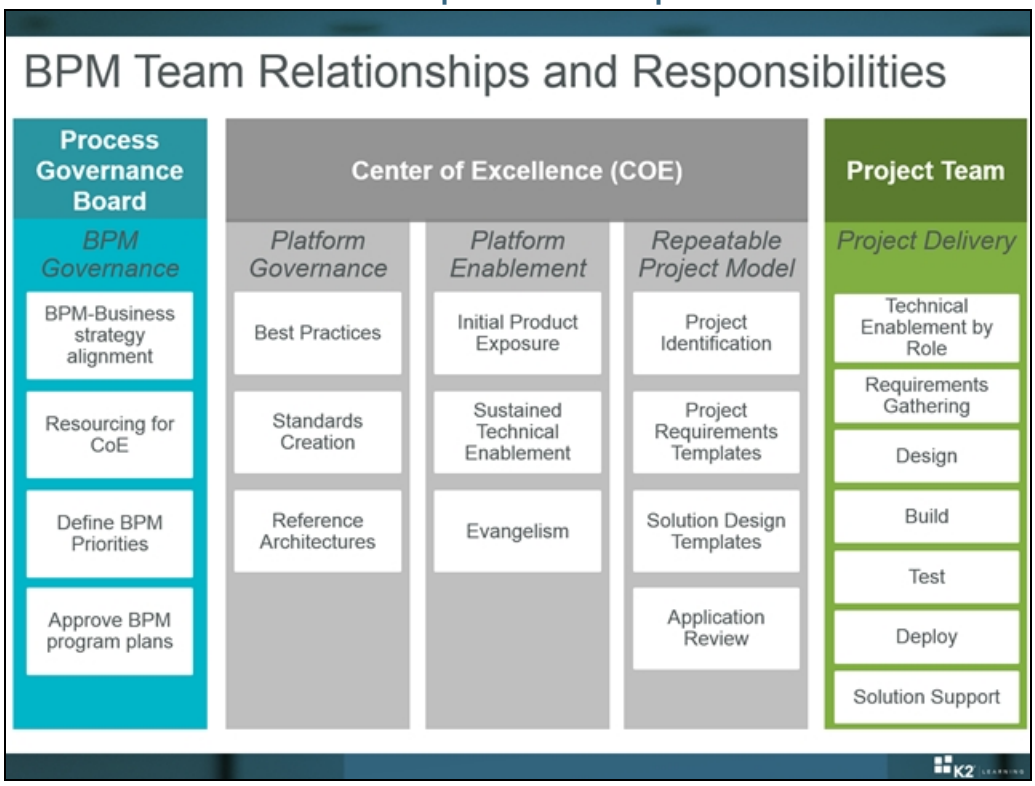


Grow

- Platform evangelism
- Community networks



BPM Team Relationships and Responsibilities



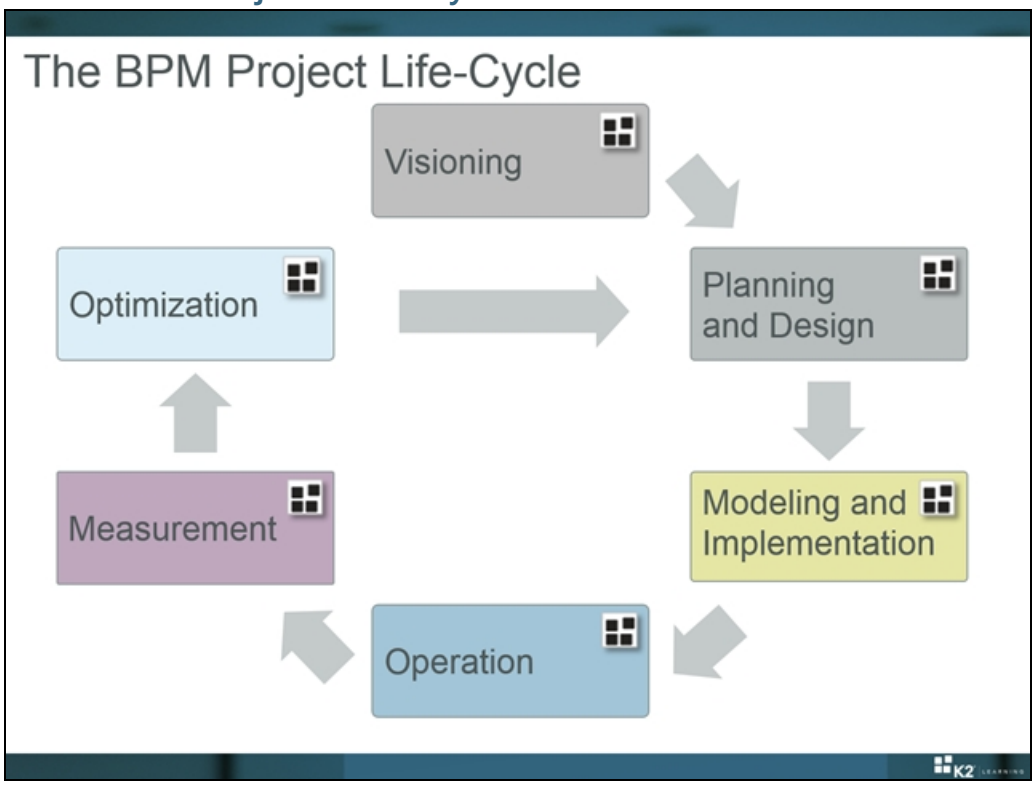
When a CoE is established, it is a good idea to clarify the responsibilities of the various teams that are involved in a BPM implementation.

The Process Governance Board's primary responsibility is to enforce BPM governance by aligning the organizational strategy with BPM, allocating resources for the CoE and approving BPM project plans. Essentially, they make sure that the CoE can execute on a BPM strategy.

The CoE is primarily responsible for Platform Governance (in other words, making sure the platform is used correctly), enablement on the BPM platform (this may or may not include initial training, but definitely should cover ongoing enablement as the platform grows over time) and ensuring a repeatable project model is followed.

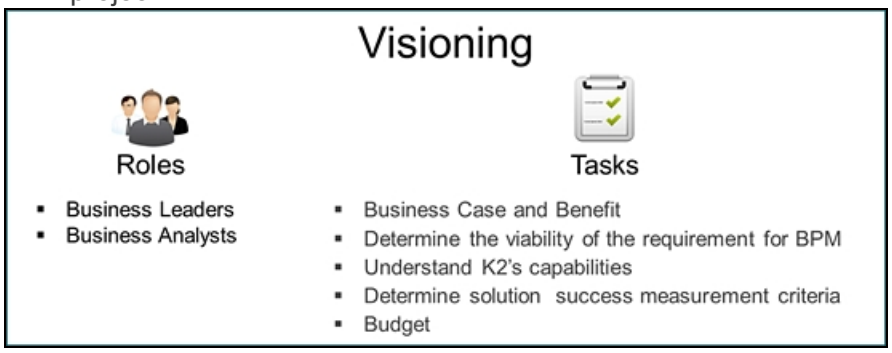
The project team is responsible for the Project Delivery, which includes role-level enablement and the design, build, test, and deployment stages. The CoE supports the project team during a delivery and provides guidance and governance, but the project team has to build the solution.

The BPM Project Life-Cycle

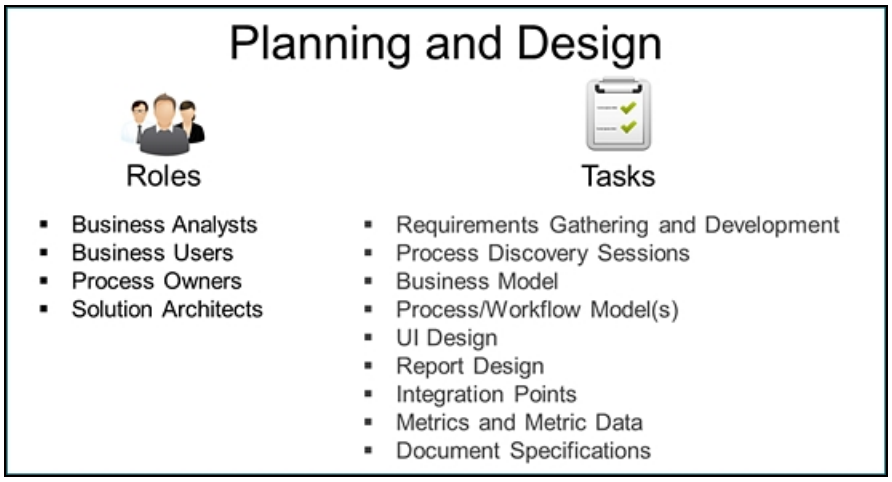


Let's break down a typical BPM project life-cycle into the various phases we usually see:

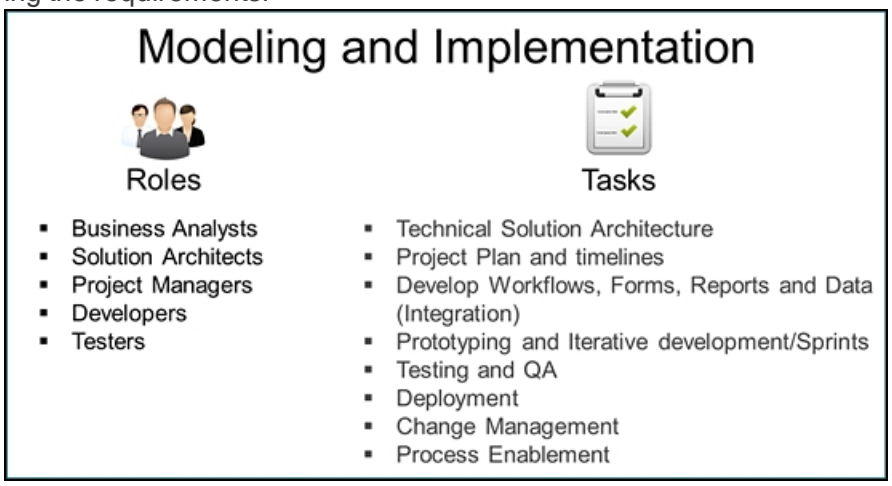
1. **Visioning** - this is usually a one-time event and the entry point for a project. Here we establish the purpose of the BPM project.



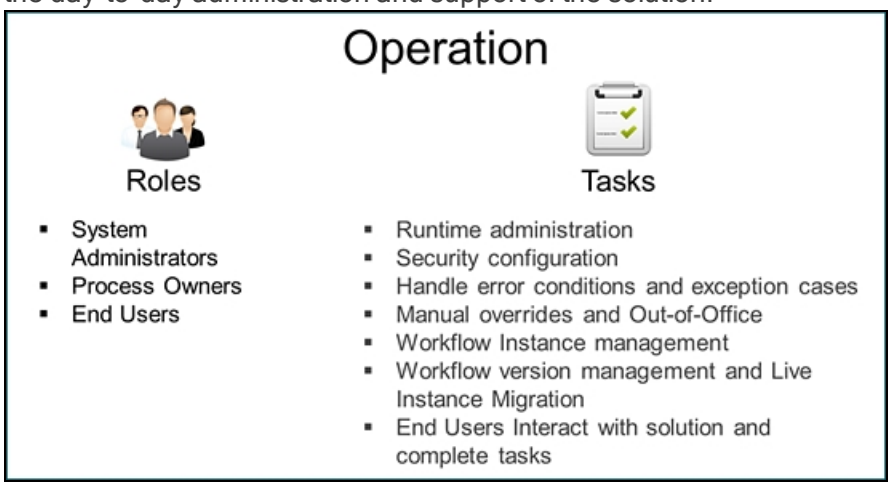
2. **Planning and Design** - this is where the project is designed, requirements and specifications are gathered and documented and the documentation is generated for the modeling and implementation phase of the life-cycle.



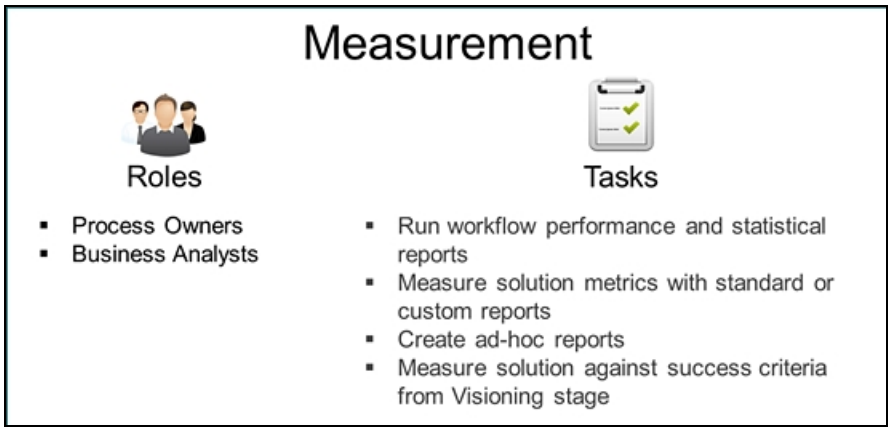
3. **Modeling and Implementation** - this is when the project is implement by the developers/designers. This may include prototyping, iterative development cycles of the actual solution components, testing and QA and deployment planning. The development process is usually iterative so ensure that the solution components are meeting the requirements.



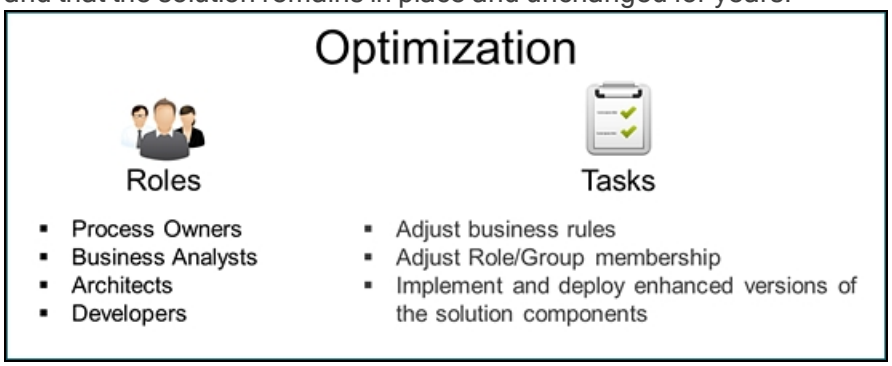
4. **Operation** - once the project is deployed, there may be administration and maintenance tasks. Think of this as the day-to-day administration and support of the solution.



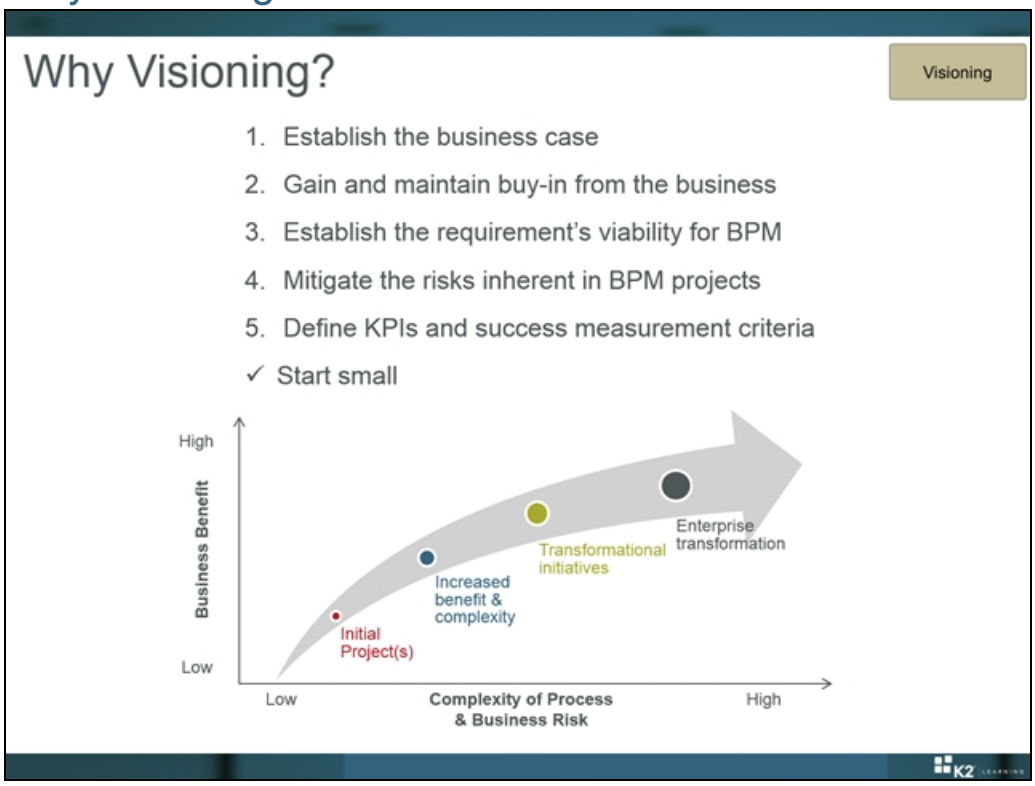
5. **Measurement** - it is important to measure the success of the project against metrics and expected benefits. In this phase, standard and custom reports are used to run BI on the solution to see how it is doing.



6. **Optimization** – the project team can start to optimize the solution based on the results of the measurement. If areas for improvement are identified, we start again with Planning and Design based on the optimized approach. (It is not necessary to start with visioning again). Usually, these subsequent iterations go faster since there are usually on minor changes or tweaks required to the solution, but it is important to note that the solution will usually evolve with the business over time. It is very rare that a BPM project is developed and implemented once-off and that the solution remains in place and unchanged for years.



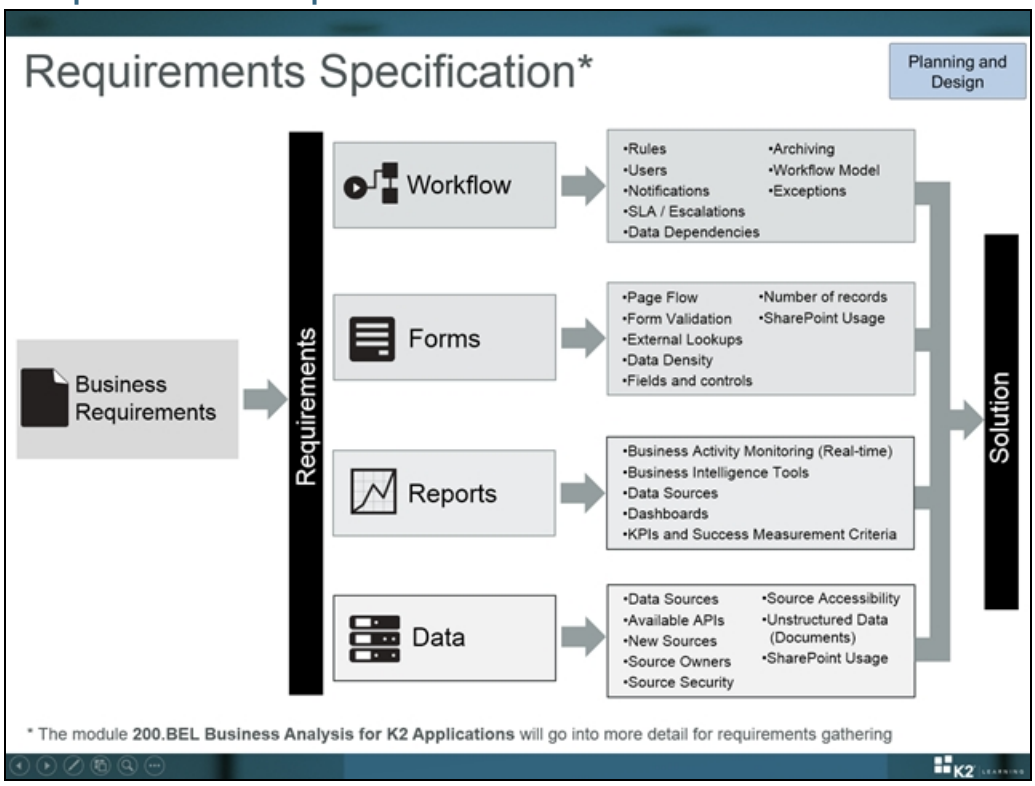
Why Visioning?



Visioning is important to put the foundation in place for a successful project. This is when you gain buy-in and support from the business for the project, and define the measurement criteria that will help to measure the project's success. Basically, visioning makes sure that there is a real business case for the project and that it suits the organizational strategy.

Note that BPM is a transformational exercise and as such, it is a good idea to start with the smaller, initial projects before trying to dive into enterprise transformation. Typically, enterprises start with smaller projects first and mature to larger and more transformational projects later on. This is in line with the BPM maturity phases that we covered earlier: as the business' BPM strategy matures, larger and larger projects are implemented.

Requirements Specification*



During the planning and design stages, the main principle is to gather specifications for Workflow, Forms, Data, and Reports that will eventually feed into the solution build phase. Because of the way the K2 platform components work together and can be re-used, it is a good idea to separate requirements gathering into Workflow, Forms, Data, and Reports.

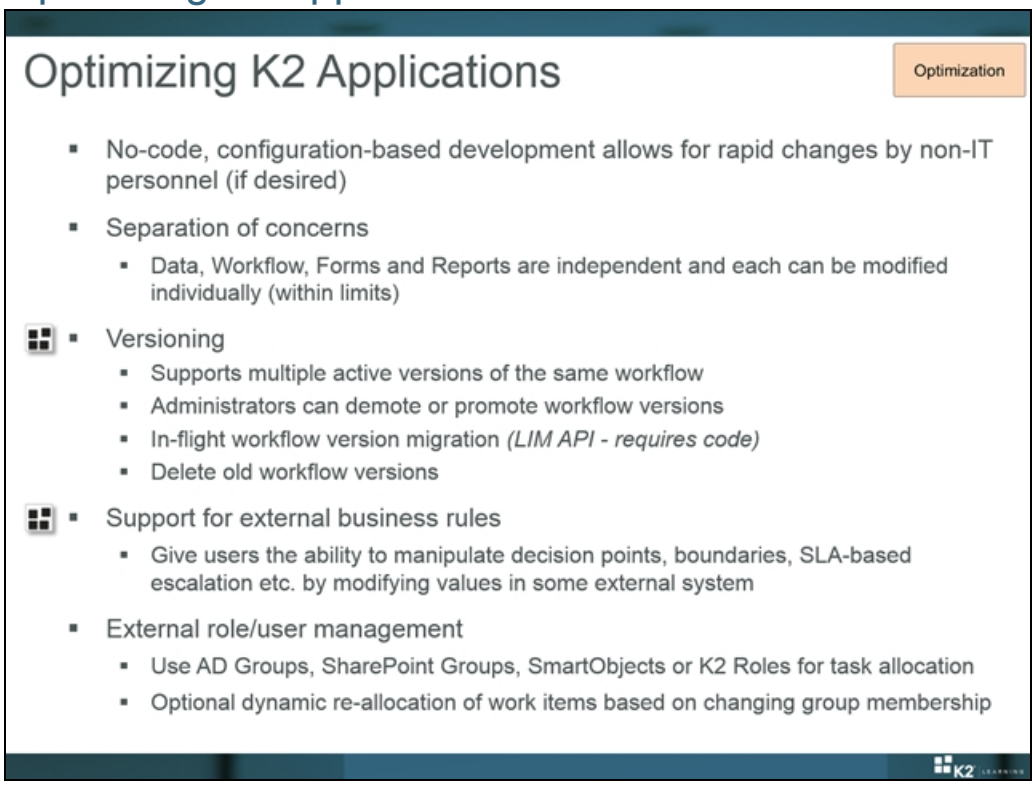
The 200.BEL course module, offers you templates to help you gather your organization's needs into Workflows, Forms, Data, and Reports. In this diagram for example, notice that the Workflow block includes the kinds of information you will need to gather to build workflows:

- Rules
- Users
- Notifications
- SLA/escalations
- Data dependencies
- Archiving
- Workflow model
- Exceptions

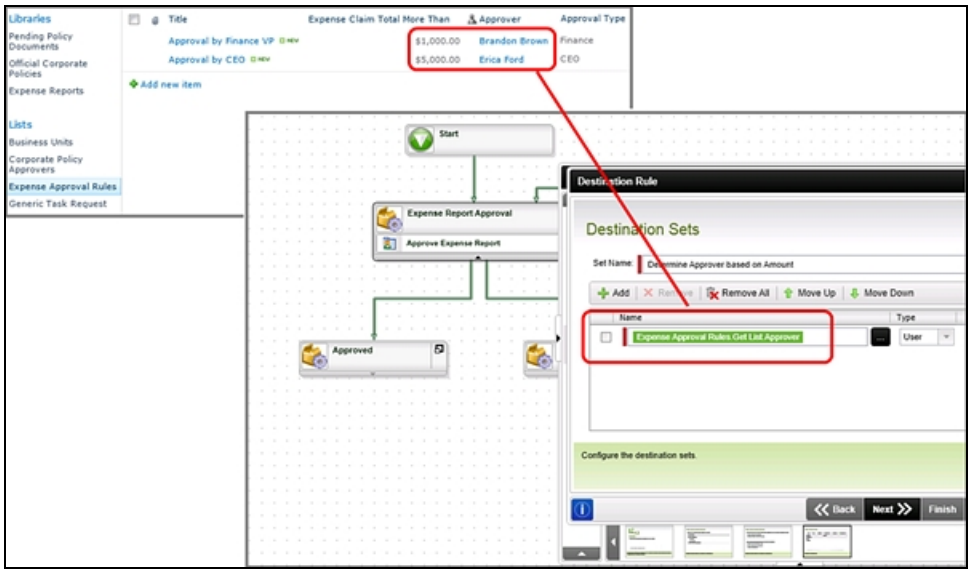
We will not delve into each of these requirements here, but simply use them to illustrate the kinds of information that you will need to gather when designing and preparing to develop workflow solutions.

*For more information on requirements and design, see module 200.BEL Business Analysis for K2 Applications.

Optimizing K2 Applications



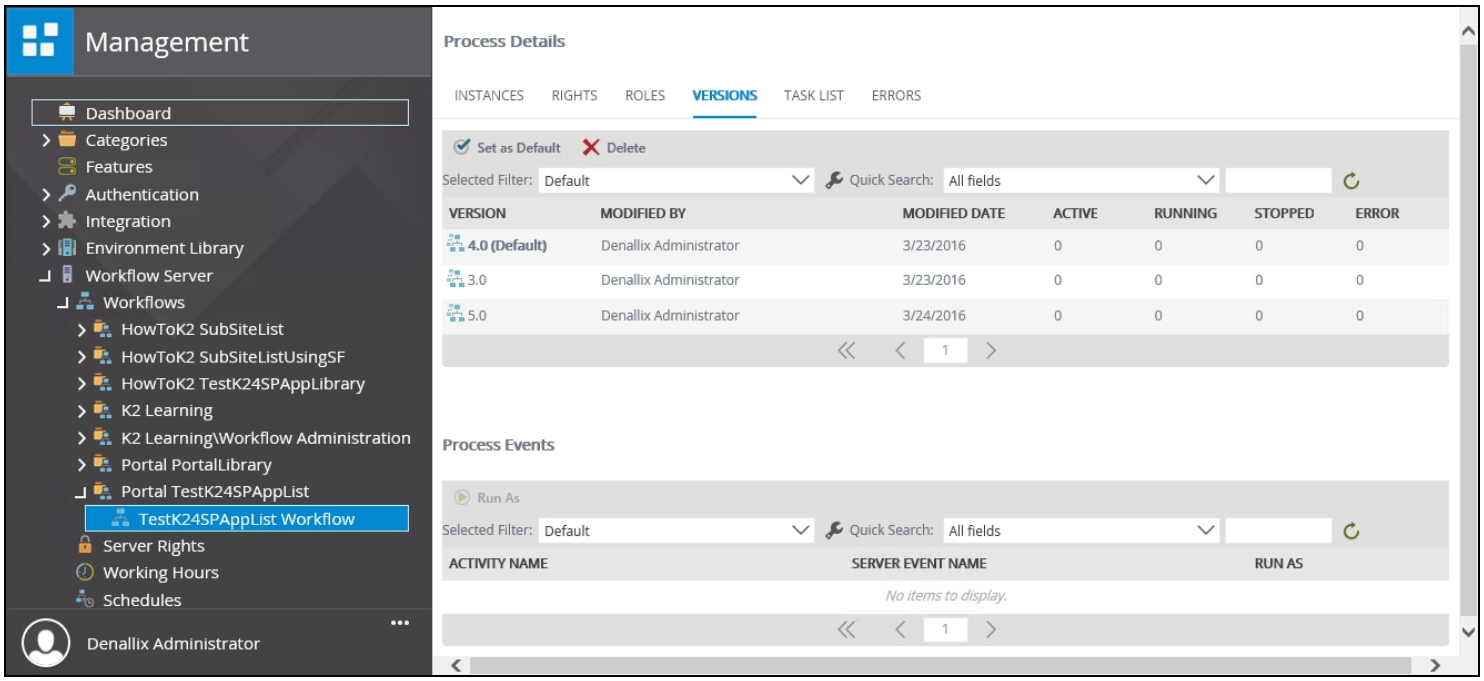
Optimizing K2 applications depends a lot on how the solution was implemented to begin with. If business rules were externalized, it can be as simple as allowing business to administer those rules without requiring any IT involvement. The K2 design tools are visual and configuration-based, so it is possible for non-technical designers to optimize the solution components if necessary. The components are relatively separate, which means that it is possible to modify Forms without affecting the underlying workflow, or modifying data sources without affecting the Forms and Workflows that use those data sources. There are some limits to the separation of concerns though, and these are discussed further in the learning modules that deal with implementing solutions in K2.



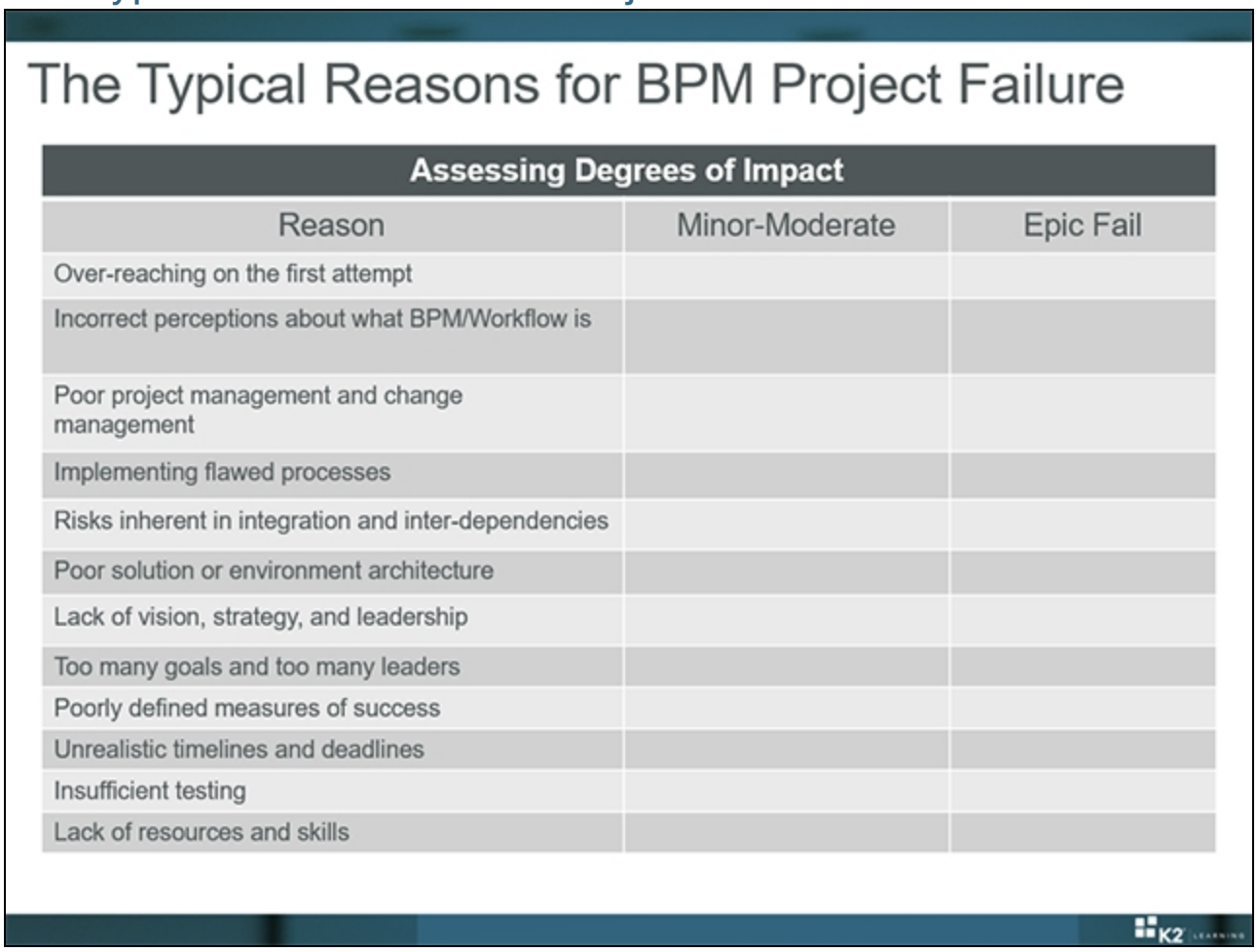
Versioning

A short note on workflow versioning: K2 supports the concept of multiple versions of a workflow, and it is possible that multiple versions of the workflow may be active at the same time. To simplify things somewhat, a workflow instance will start on the Default version of the workflow (usually the latest version that was published) and the workflow will continue running on that version until the workflow completes, even if a newer version is published in the meantime.

It is possible to migrate in-flight workflows from one version to another, and that will be discussed in another learning module. The example below shows how a K2 admin has used K2 Management to change workflow version 4.0 to the default version.



The Typical Reasons for BPM Project Failure



Let's take a look at some of the typical reasons why BPM projects experience a “lack of success”. Clearly the list of reasons shown in the table above is only a **partial** list of reasons BPM projects fail. There are too many possible reasons and combinations of reasons to mention here. But seldom does a project fail as a result of **only one** of the reasons shown here.

This table gives us a way of not only examining the common reasons projects fail, but it also provides a way to evaluate the impact of those factors. Minor to moderate impacts are those that may:

- Cause misunderstanding or conflict among team members
- Contribute to poor team morale
- Delay or add costs to a project
- Compromise future projects
- Produce inefficient processes

On the other hand "epic fail" factors are exceptionally costly or catastrophic to the company, its mission, its morale, its products, or its financial future. Think of these as unchecked or serious misjudgments that lead to terminations.

With this distinction in mind, look over the table above and decide how much impact each factor or combination of factors has. By itself one factor may only have a minor to moderate impact, but combined with others, the resulting failure may be truly "epic." You can base your decisions on your own experience or actual examples you are aware of.

Understanding these contributing factors and assessing their severity can help you recognize and mitigate some of these risks with a structured and repeatable Business Analysis.

Ensuring Success in BPM Projects

You may recall some of the risk factors from the previous slide that contribute to BPM project failure. Now look over the possible strategies listed here and see if you can match a mitigating strategy with one of the problems from the BPM failures list on the previous slide. Consider the following example.

Risk factor

Over-reaching on the first attempt

Description

Trying to implement too much on the first attempt at a BPM or workflow project.

Example

There is often a desire to prove a new BPM approach or product by applying it to a large, complex requirement.

Mitigating Strategy

Select the right "first process". (see item 2 above)

Questions to Ask

What style of workflow or process is this?

Can the process be formalized?

Is this a structured or unstructured process?

Can the process be separated into smaller workflows?

What are the main goals of automating this process?

How does this process fit into the bigger picture?

Which systems are involved in the process?

Who is involved in this process?

Is there a clear set of requirements, deliverables, and timelines?

Now let's look at the other strategies shown here.

Understand why the project is important to the organization

Why is all this important? Because if the project does not support any of these fundamental organizational aspects, the chance of project failure is very high. Projects that do not support the Strategy or do not support or enforce the Core Values usually fail due to lack of support from higher levels and/or lack of support from the ground up. To make sure a project is successful, it is critical to demonstrate that the solution has value to the organization because it supports the Strategy and Core Values, and therefore will help the organization reach its Vision.

Remember the 80/20 principle

This rule, known as the Pareto principle, has application to ensure success in BPM projects. Trying to reach 100% satisfaction and 100% compliance to every single requirement identified at the envisioning or discovery phase almost guarantees a 100% chance of failure. Try instead to implement around 80% of the requirements in the first iteration, and then work on the remaining requirements as needed. Chances are the remaining 20% of requirements are not that important.

Gain and maintain buy-in from the business

- Involve the users
- Include the detractors
- Demonstrate value to the strategy and vision
- Collaborative process discovery sessions
- Manage process discovery sessions
- Know when to say “yes” and when to say “no”
- Use prototypes
- Let the users play early-on
- Maintain executive sponsorship
- Create a steering group

Manage development and testing cycles

A well-managed and executed development cycle is vital to ensure successful BPM and workflow projects. Specific challenges can arise in BPM projects just due to the nature of the challenge being solved or the scale of the proposed solution. This most commonly manifests itself as fluctuating or vague requirements and specifications, and an increased risk of scope creep.

Here are a few approaches that can be applied during the development and testing cycles of BPM/Workflow projects to help mitigate and respond to these risks.

- Manage the specifications and requirements
- Use the available K2 Resources
- Prototype early
- Use iterative development
- Manage “scope creep”
- Design for integration and scalability
- Design for control
- Enforce development standards and practices
- Enforce consistent testing approaches

Promote and support user adoption and business ownership

User Adoption

Design and build the application correctly

This is an obvious point. A poorly designed and implemented solution has little chance of user acceptance. There are a few common techniques and considerations you should bear in mind when designing and building the solution to ensure user adoption.

- Solicit input
- Make it easy to use
- Make it transparent

Use Change Management techniques

- Focus on the positive
- Communicate the value

Demonstrate the relative and tangible advantage of the solution

Ensure compatibility of the solution with the strategy

Reduce the complexity of the solution

Give users a chance to trial the solution

Find and leverage the internal evangelists to increase visibility

Business Ownership

Apart from user acceptance, it is also important that business is willing to take control of the solution. You do not want the development team to have to respond to first-level support issues, or to have them constantly tweak and adjust the process to meet changing requirements. In many cases, a solution may have been created by a few different contractors who may leave the project after completion. In this case, it is especially important that the business will take ownership of the process.

Build the solution for control, not restriction

Communicate the value

Delegate the day-to-day management

Measure success with KPIs, ROI, metrics, adoption, and feedback

KPIs

The best way of establishing concrete measurements is to define a set of Key Performance Indicators (KPIs) that support the goals and vision of the business while still being measurable.

ROI

Return on Investment is a tricky measurement to apply against BPM processes due to the sheer size and scope of many BPM projects.

Project metrics

Ideally, BPM project should have hard metrics that can also be used to measure the success of a BPM implementation.

Adoption and user feedback

Ultimately, adoption of the process is a critical measure to the success of the project. Even if all your project metrics are met it means nothing of the process or solution is not used.

User feedback is a softer measurement that can also help to gauge the success of a BPM project. The best way of measuring user feedback is to distribute questionnaires with rating-style questions to the actual users of the solution, and then averaging out the responses.

Create a Center of Excellence (CoE)

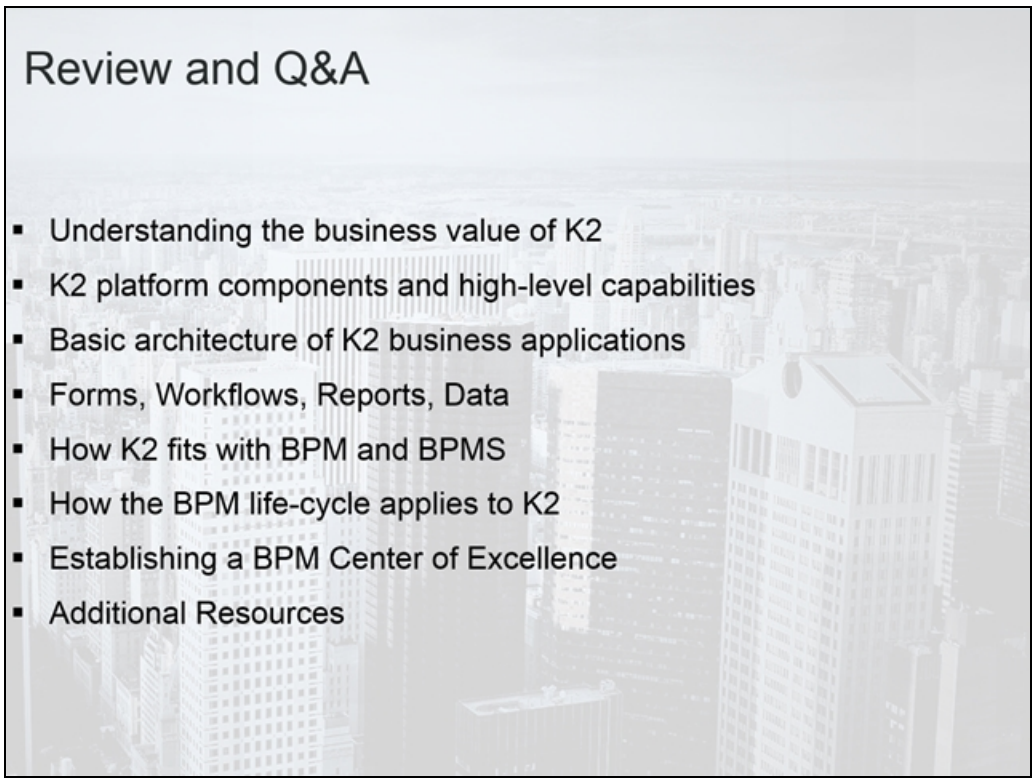
Creating a Center of Excellence (CoE) is vital when moving to wider adoption of BPM in an organization. There are two main goals of establishing a CoE:

1. A CoE helps the business understand, apply, and achieve the benefits of BPM and BPA
2. A CoE helps project teams to use the BPM framework correctly and effectively and conform to guidelines and standards

A CoE is mostly a collection of resources and one or more people that would help to achieve these two goals.

This is just a partial list of possible strategies that can help mitigate the risks in a BPM or workflow project. The most common risks can be addressed easily with specific approaches and strategies. What is most important is that an organization understands these risks and takes the necessary preventative strategies to prevent these factors from affecting its BPM projects.

Review and Q&A



This module was intended to give you an overview of the K2 platform and K2 business applications. You should also understand how the platform is used to address BPM and BPA requirements, and how the K2 platform components are used in a typical SDLC spanning design, assembly, deployment, execution and monitoring. Let's review the main topics covered in this module.

- Understanding the business value of K2
- K2 platform components and high-level capabilities
- Basic architecture of K2 business applications
- Data, Workflows, Forms, Reports
- How K2 fits with BPM and BPMS
- How the BPM life-cycle applies to K2
- Establishing a BPM Center of Excellence
- Additional Resources

By now you have an overall, high-level understanding of how the K2 blackpearl platform is used in an organization.

This is also your opportunity to ask some questions about the information covered in this module. Note that other learning modules will cover the various components and actual implementation of K2 in more detail, so your instructor may defer questions until a later module.

100.BHX: Introduction to K2 Applications with K2 Designer

Video

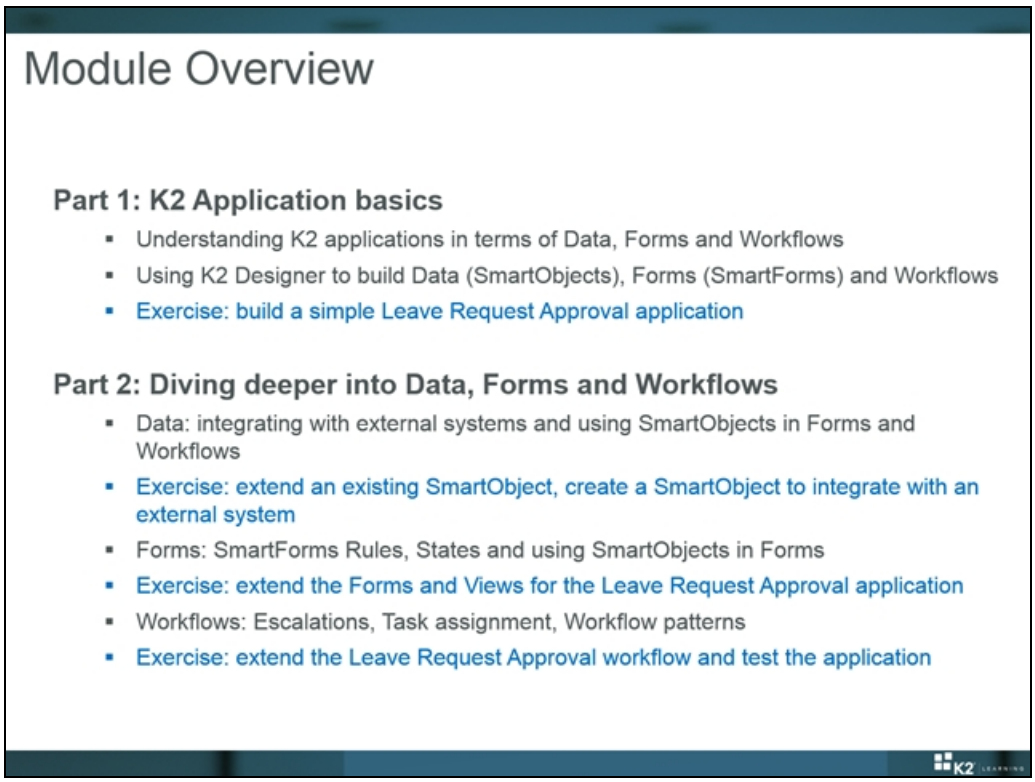
The *100.BHX: Introduction to K2 Applications with K2 Designer* training module explains how to build K2 applications in terms of Data, Forms and Workflows. In this module, we will use K2 Designer to build an application with Data (SmartObjects), Forms (SmartForms) and Workflow.

This module covers the following concepts:

- Using K2 Designer to build simple applications
- How Data, Forms, and Workflows are used to build an application
- Integrating with external systems with SmartObjects
- Integrating with external systems with Workflow wizards
- Using SmartObjects in Forms and Workflows
- Workflow concepts: escalations, task slots, workflow patterns

Note
Although this module focuses on the web-based K2 Designer which may or may not be used by all organizations, we will be explaining some fundamental concepts about the elements that make up an application, and the knowledge gained in this module will set the scene for other learning modules that delve further into Workflow and SmartObjects. Therefore, even if you do not intend to use K2 Designer in your organization, we recommend that you complete this learning module to build up your initial knowledge of K2 applications.

Module Overview

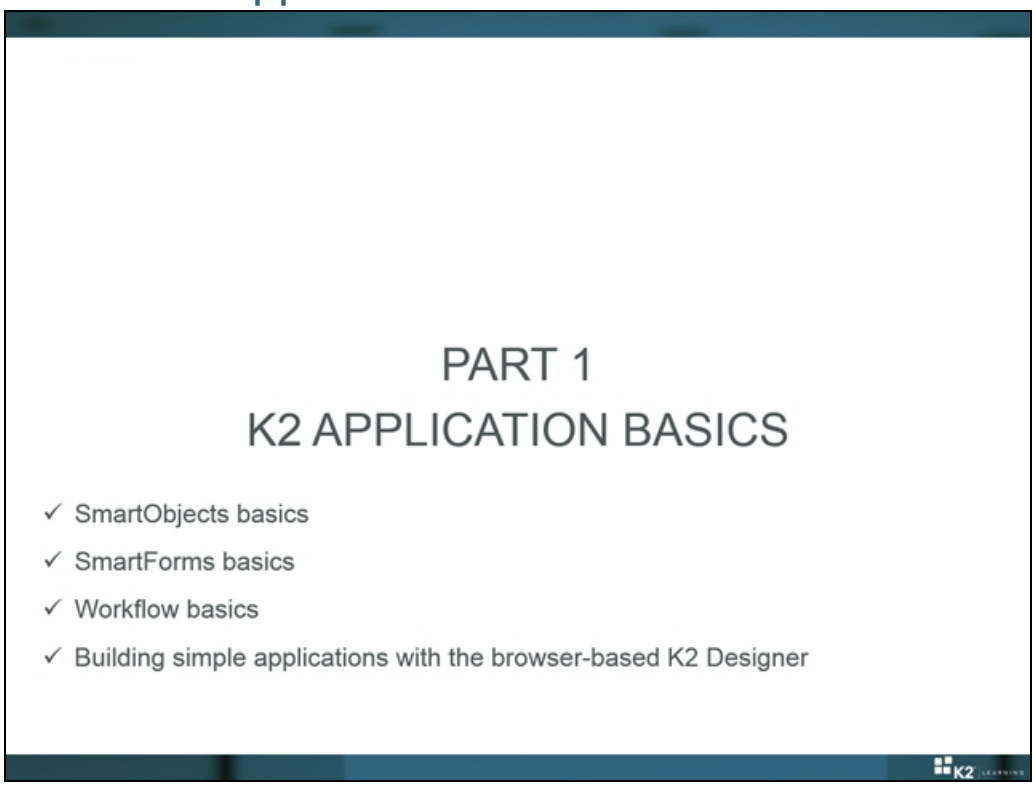


Video

This learning module consists of two parts:

- Part 1 focuses on the relationship between application elements (Data, Forms, Workflow) and how you build these in K2 Designer
  - Part 1 ends with a lab exercise where participants will build a simple end-to-end Leave Request Approval application
- Part 2 dives a little deeper into each application element. Each section has an exercise where participants will apply the concepts covered for that item to the basic Leave Request Approval application that was created in Part 1.
  - SmartObjects (integrating with external systems)
  - Forms (editing Views and Forms)
  - Workflows (extending the workflow to include rework, escalations, customized emails)

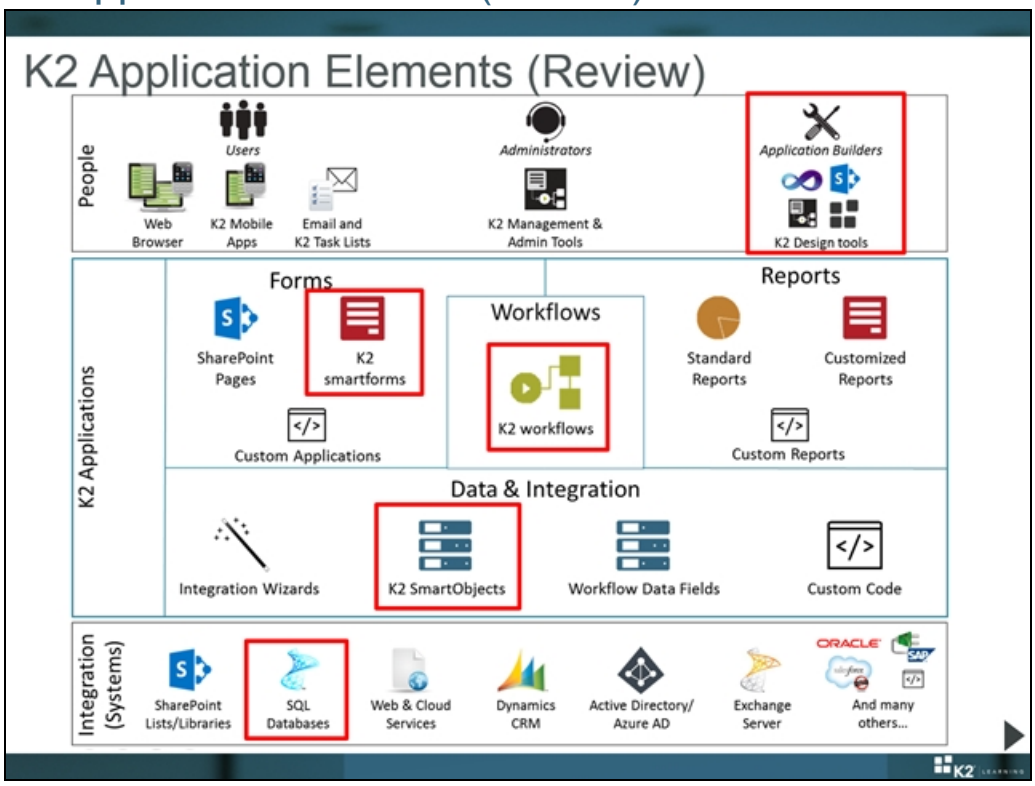
Part 1: K2 Application Basics



Video

In Part 1 we will look at the basics of K2 applications that combine Data (SmartObjects), Forms (SmartForms) and Workflows. We will be using the browser-based K2 Designer for this module. At the end of Part 1, we will build a simple Leave Request Approval application from start to end. In Part 2, we will extend upon this basic version of the application with more advanced Data, Forms and Workflows.

K2 Application Elements (Review)



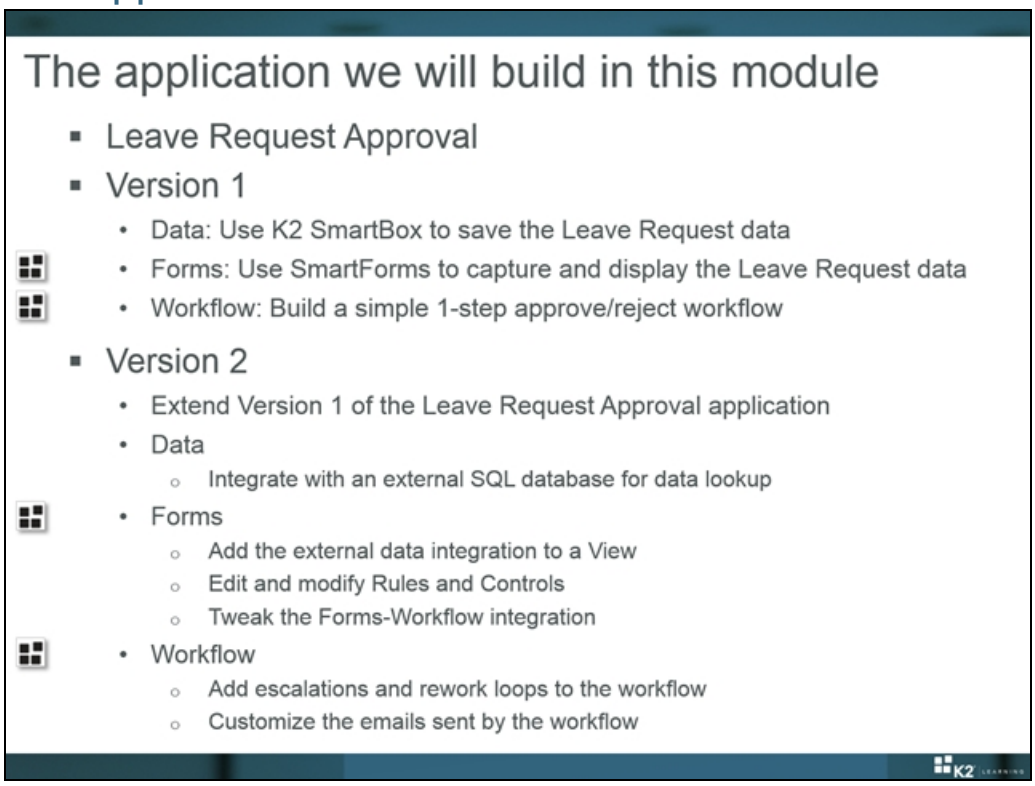
Video

From the previous learning module, you should recall that K2 applications are made up of Workflows, Forms, Data and Reports. You should also recall that these elements are designed using one of the available K2 Design tools, and that K2 uses a technology called SmartObjects to integrate with other systems.

In this particular module, we will specifically look at using the following technologies to assemble a K2 application:

- For Design tools, we will be using the browser-based K2 Designer environment to build SmartObjects, Forms and Workflows.
- For Forms, we will describe the use of K2 SmartForms for the Forms (User Interfaces) component on a high level.
- For Workflows, we will use the browser-based workflow design tool to build a K2 Workflow.
- For Data, we will build both a simple K2 SmartBox-based SmartObject, and later on another SmartObject that integrates with an external SQL database for data retrieval.
- For Reports, we will use the standard K2 workflow reports.

The application we will build in this module



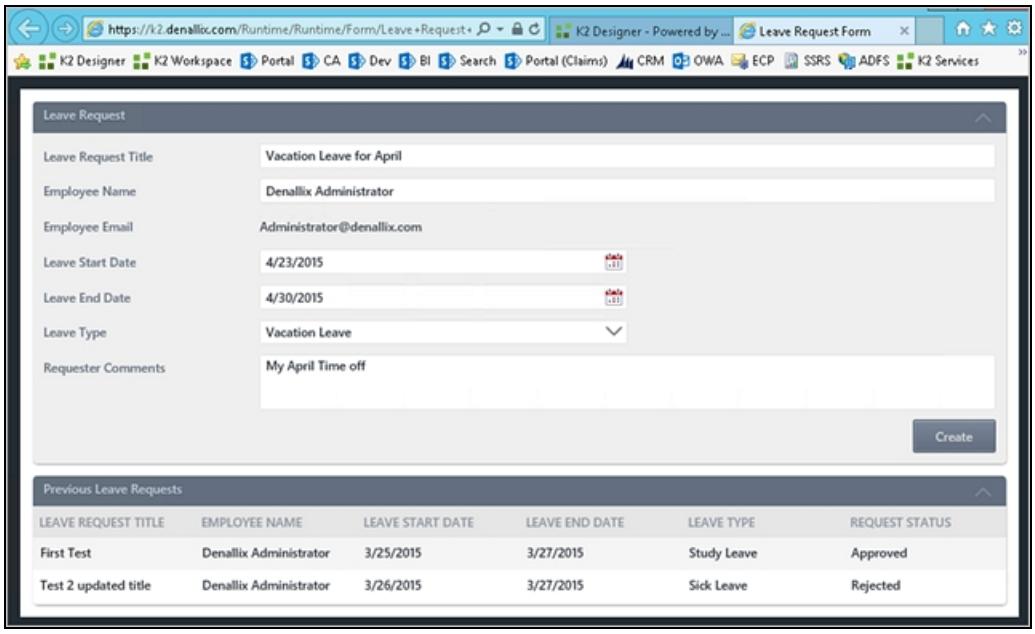
Video

The best way to learn how to use K2 is to build an actual application. In this module, we will be building a "Leave Request Approval" business application, to allow users to enter and submit a leave request which will then go through their manager for approval. Although this is a very simple application, building it will get you familiar with how SmartObjects, Forms and Workflows are used to assemble K2 applications. We intentionally do not want to create a very complex application, we are keeping things simple for this module. Later on, you will learn how to build more advanced SmartObjects, Workflows and SmartForms.

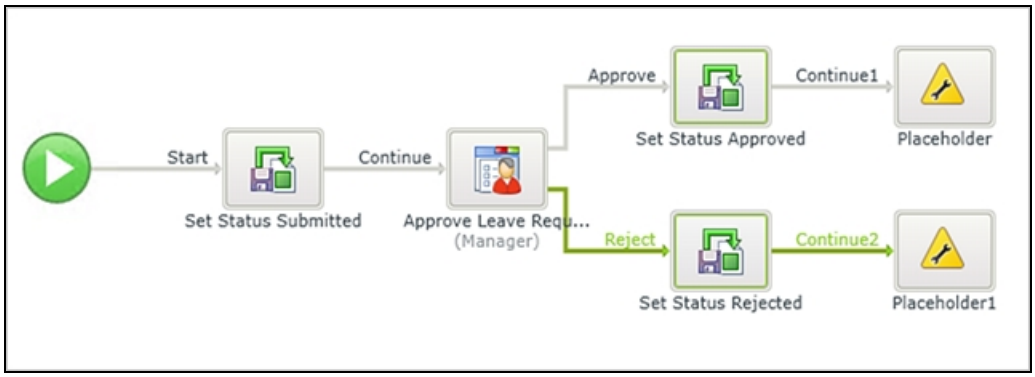
Although we use K2 Designer for this module, the concepts are the same for other K2 designers since each application will have Data, Forms (User Interfaces) and Workflows, even if you use something other than SmartForms or other K2 design tools.

We will break the application build up into V1 (Basic, simple) and V2 (Extended, more complex). Version 1 will not have any integration with any external systems, it will be based purely on K2 only. For the first version, we will build the entire application in one exercise so that you can see how the components “plug” into each other.

The below screenshot shows an example of the Form that we will build in Version 1 of the application.



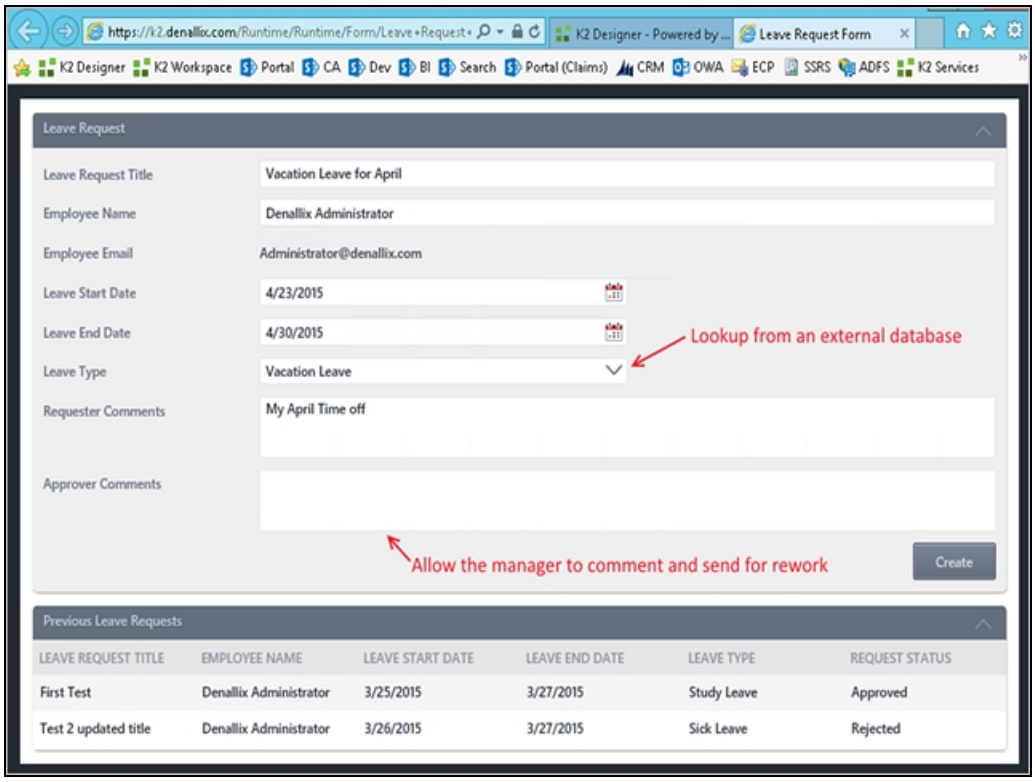
And this is what the first version of the workflow will look like:



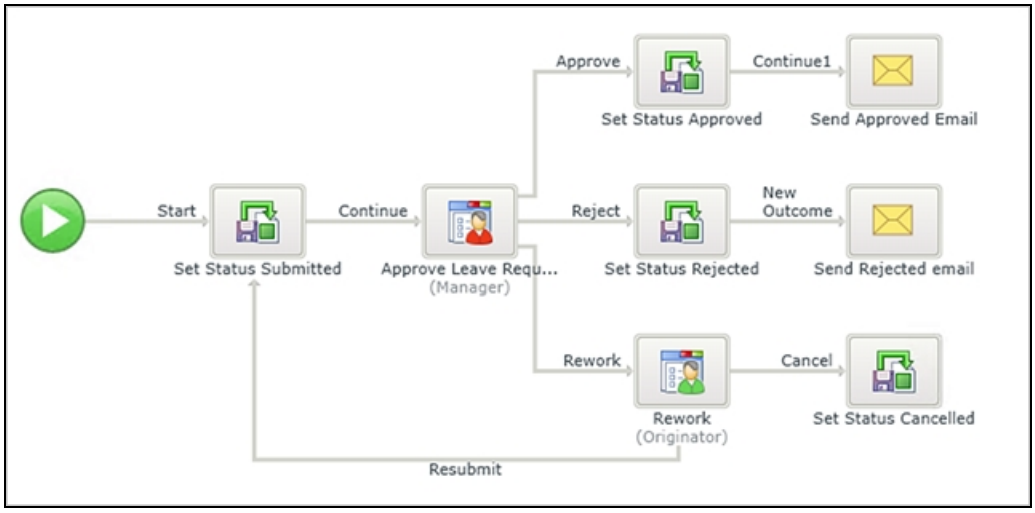
For the second version of the application, we will look a little deeper into the application elements (Data, Forms, Workflow) and for each of the components, edit the application that we created.

For the Data component, we will edit an existing SmartObject, then create a new SmartObject to integrate with an external SQL database.

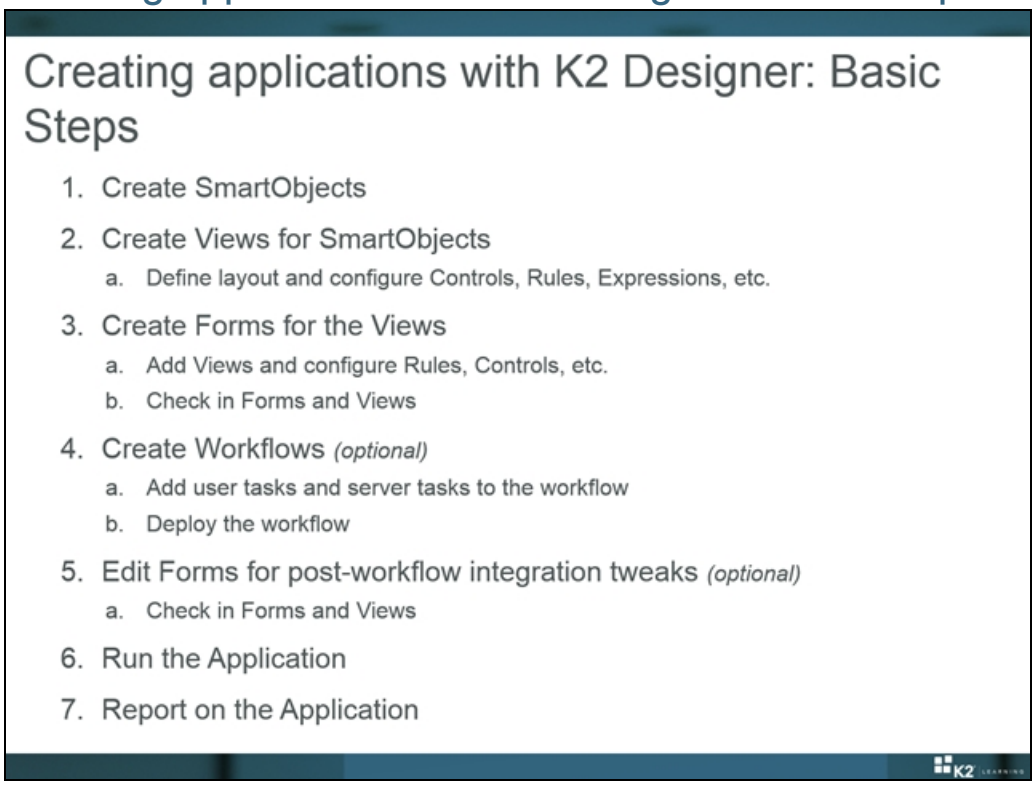
For Forms, we will edit Forms and Views to include external integration and some more Rules. The screenshot below shows "Version 2" of the Form.



For workflow, we will add escalations, customize the notification messages, build a rework loop and edit the Forms for additional workflow integration tweaks. The screenshot below shows the final "Version 2" of the workflow for our Leave Request Approval application.



Creating applications with K2 Designer: Basic Steps



Video

This slide lists the basic steps you need to follow when creating applications with K2 Designer. You don't HAVE to build applications in this order, but this is the usual approach.

We usually start with SmartObjects, then create Forms and Views for those SmartObjects, then create workflows if any are needed. By that time the application is usually ready, although you would typically go and edit the forms for the new workflow integration to make them behave differently depending on where in the workflow the forms are being used.

You have to deploy workflows before they will be available to your users. Deploying the workflow "injects" the necessary rules into the Forms and Views. You have to check in Forms and Views before the will be available to your users. (If you need to "promote" applications between environments, you would use K2 Package and Deployment tools to do so.)

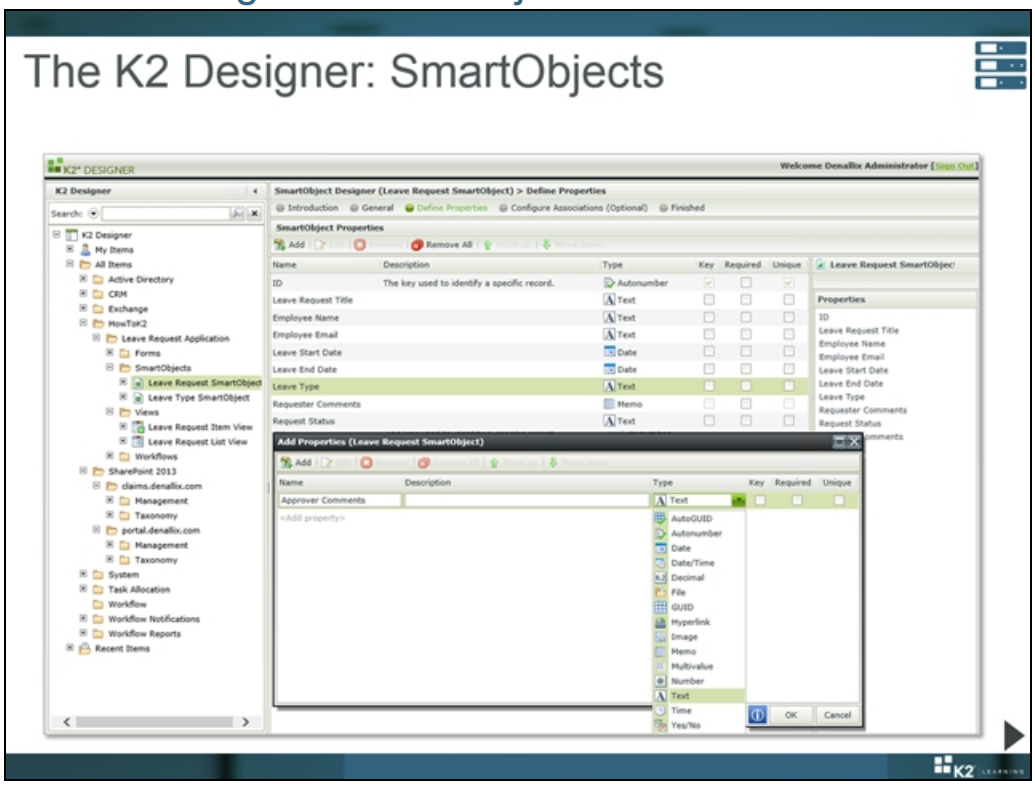
Once the application is deployed, you can run the application and report on the application.

To summarize, here are the main steps to create applications with K2 Designer:

1. Create SmartObjects using tools like K2 Designer or K2 Studio. You could also auto-generate SmartObjects in some cases to integrate with other systems like SQL databases.
2. Create Views for SmartObjects using K2 Designer. In this step you will build a user interface that exposes the SmartObjects you created in Step 1. As part of this step, you would also define the View layout, add and configure Controls, Rules and Expressions. You may need to create multiple Views for the application.
3. Create Forms for the Views. You have to add Views to Forms since users interact with Forms. In this step, you would take the Views created in step 2 and add them to a Form. You would also configure additional Rules and Controls on the Form as needed. Once this is done, check in the Forms and Views.
4. The next (optional) task is to create workflows. This is optional because not all applications have a workflow component but when they do, it is best to create the Forms used in the workflow first, before you start building the workflow. While building the workflow, you will add user tasks and server tasks to the workflow and configure them using wizards. When the workflow is complete, deploy the workflow.
5. Next, you may need to check out and edit the Forms for post-workflow integration tweaks. This is also optional, but sometimes you may want to change the way in which workflows are integrated with your Forms or make some enhancements now that the Forms are workflow-aware. If you do change the Forms, remember to check in Forms and Views when you are done.

6. Next, run the Application to test it, or ask users to start testing the application. At this point you may want to set permissions on the workflow so that a large (or limited) group of users can start the workflow.
7. Once some instances of the application's workflow have started, you can use the available reports to report on the application.

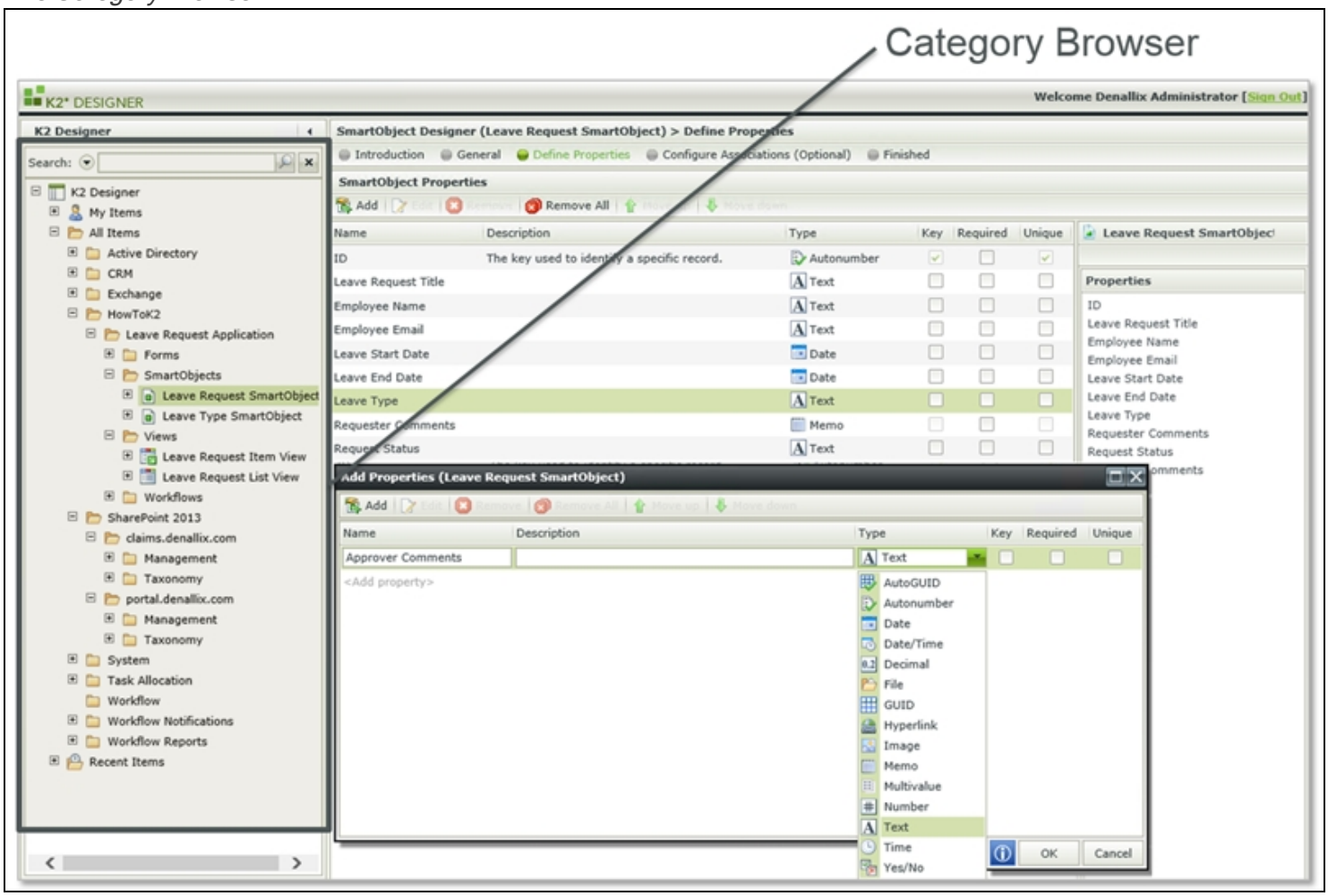
The K2 Designer: SmartObjects



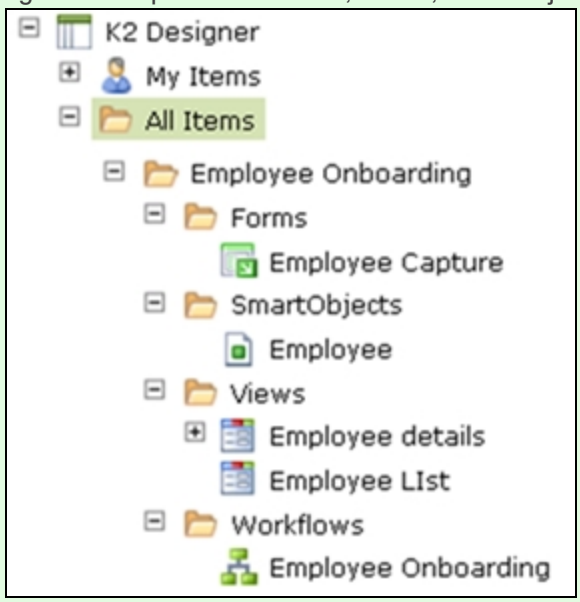
When working with K2 Designer to build SmartObjects, there are two main areas in the design canvas: The Category Browser and the Properties pane.

Think of the Category Browser as a repository where you can store the definition of the SmartObject. It works very similarly to a File System with folders that contain items. In most cases, you would create folders in the Category Browser that line up with the applications in your environment, and you may choose to create sub-folders where needed.

The Category Browser

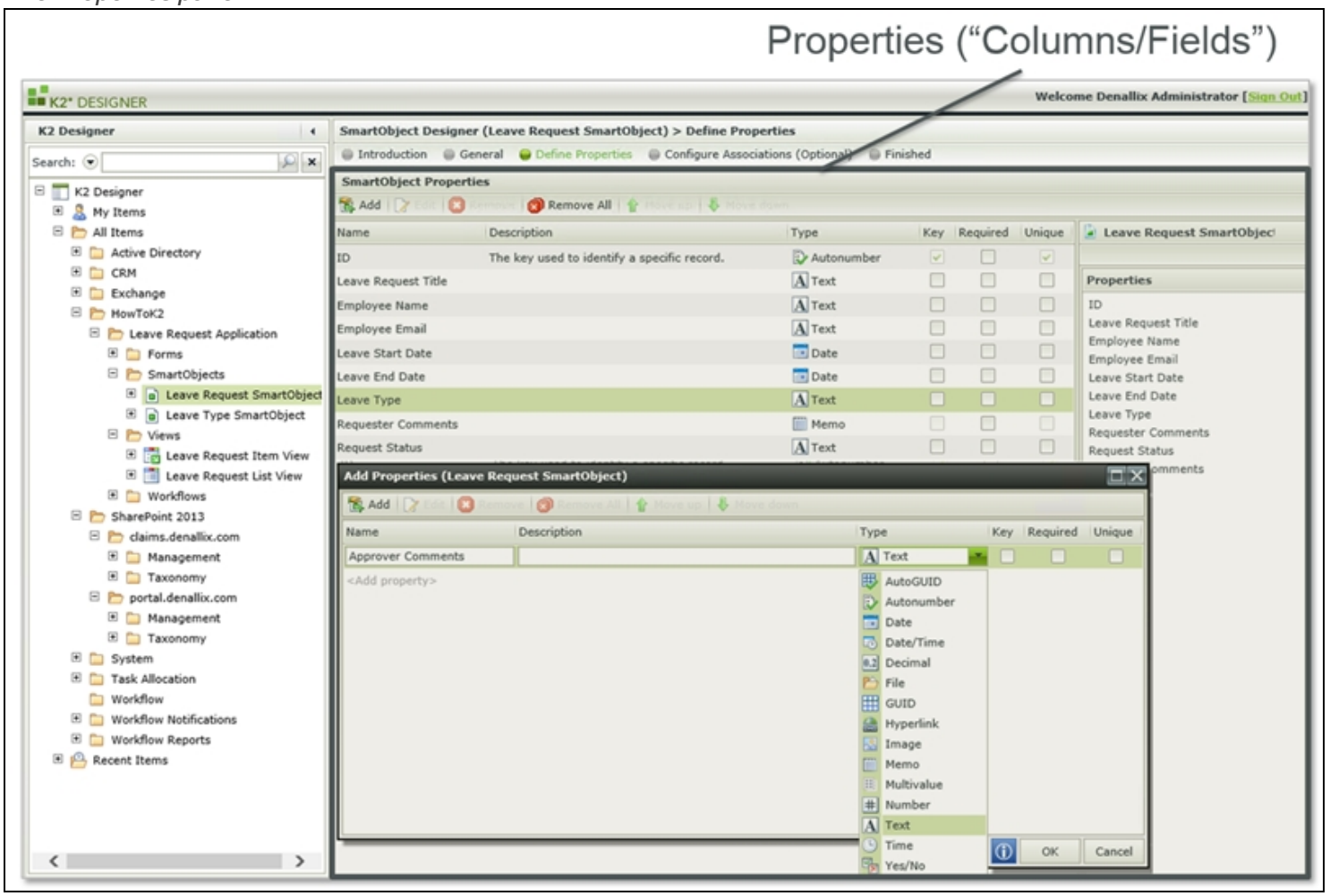


Tip
You could use sub-folders to keep all the elements (SmartObjects, Forms, Views and Workflows) grouped together. While you may choose your own categorization system, one recommended approach is to use categories to separate the Forms, Views, SmartObjects and Workflows in a solution, as shown in the example below.

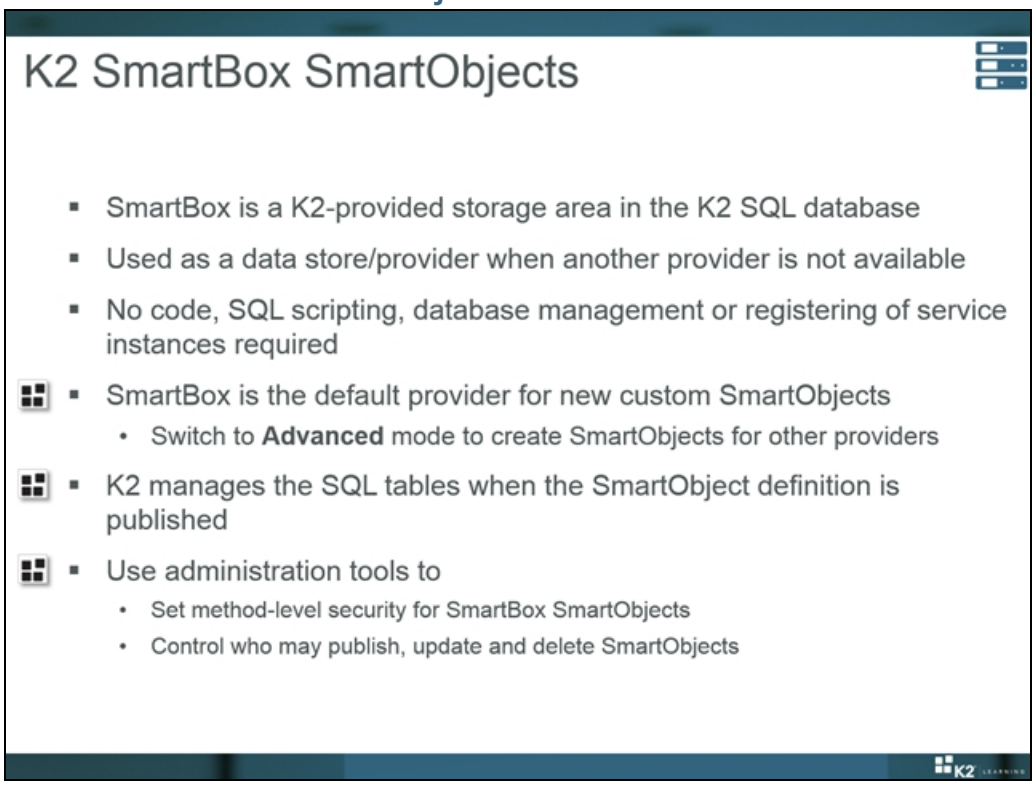


The Properties pane is where you define the "columns" or fields, that make up your SmartObject. Here you can define the fields and the data types for the fields. For advanced-mode SmartObjects (which we will look at later), this is also where you would configure the methods that point to particular back-end systems.

The Properties pane



K2 SmartBox SmartObjects

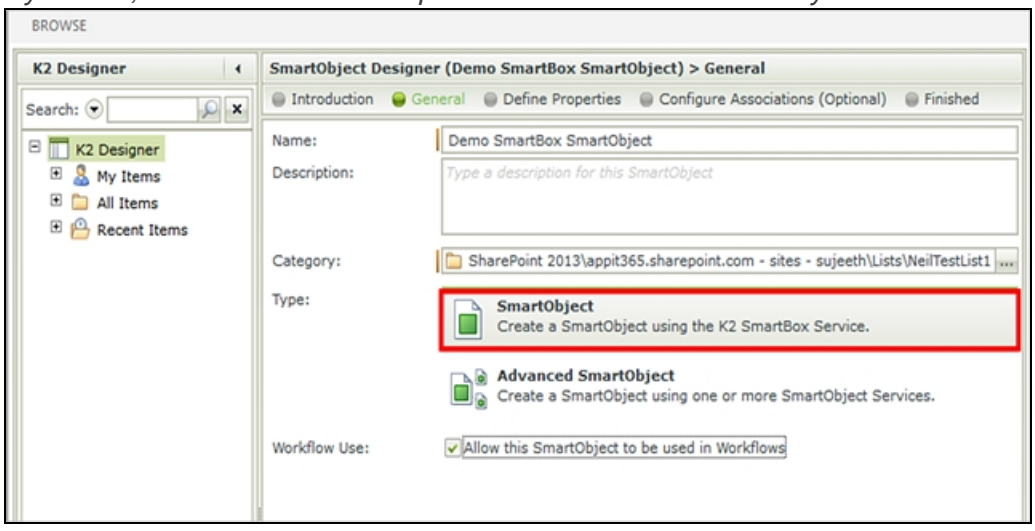


Video

SmartBox is K2-provided storage within the K2 database, used when you do not already have an underlying system to store the data. Suppose you wanted to define a collection of "Regions" so that you could use this list of Regions in a drop-down control on different forms, and suppose that your organization does not already have this list of Regions defined anywhere. While you could ask IT to create a SQL table somewhere and then expose that table using Service Instances as we described before, another approach is just to define this "Regions" list as a SmartBox SmartObject so that the data lives in the K2 database. This is a slightly easier approach because K2 takes care of creating and managing the underlying SQL tables that will store the data, all you need to do is decide what properties should exist in the SmartObject.

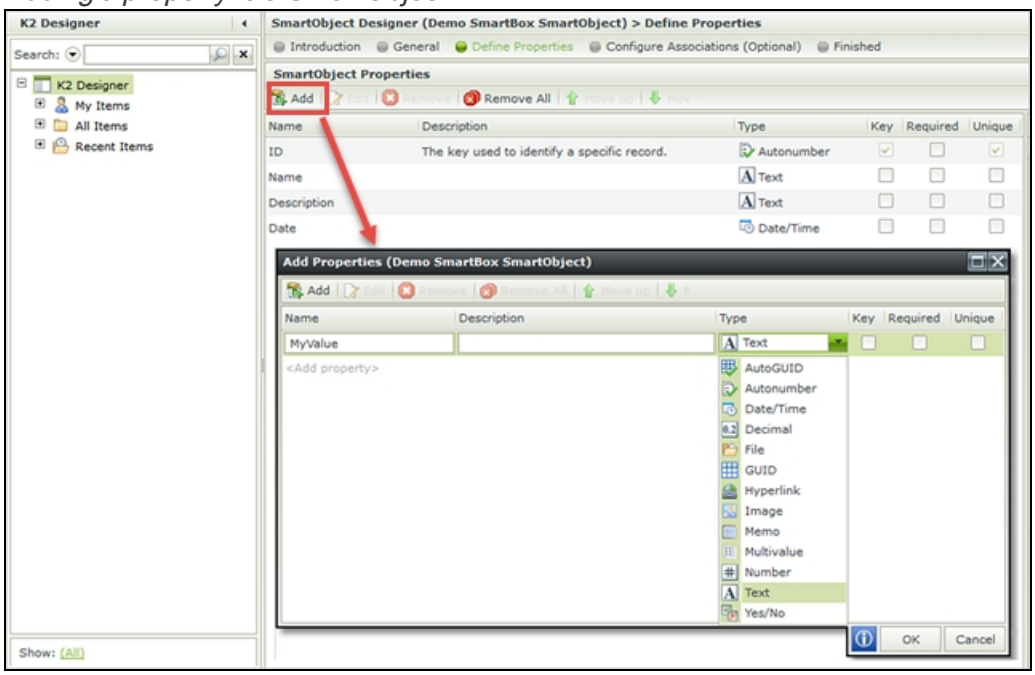
When you create a new SmartObject in K2 Designer, SmartBox is the default provider that is selected, which means that K2 assumes you want to store the data in the SmartBox. If you want to retrieve data from another system, use the Advanced SmartObject option. Selecting the **Allow this SmartObject to be used in Workflows** option will expose the SmartObject to workflows designed with K2 Designer.

By default, SmartBox is the Data provider that will be used when you create a new SmartObject

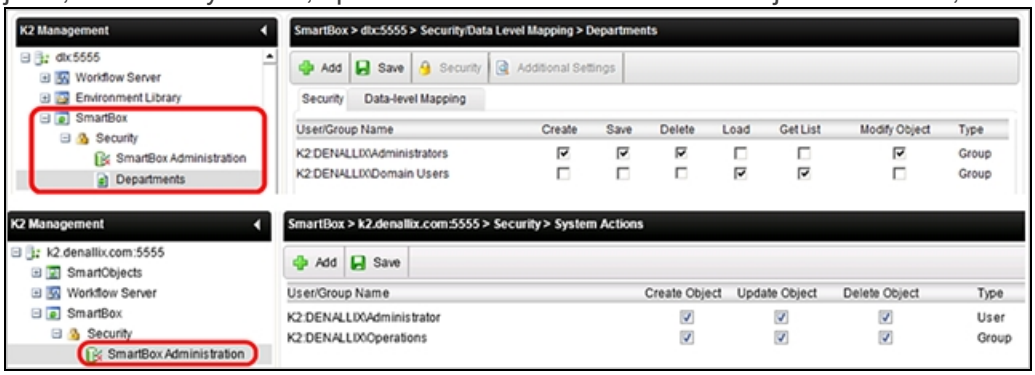


Next you will add Properties to your SmartBox SmartObject and choose the appropriate data type for each Property. By default, K2 will include the standard Create, Read, Update, Delete and List methods used to access this data; you can remove some of these if desired. When you are done, publish the SmartObject and then K2 will create the underlying data tables for you. Once the SmartObject is published you can generate Views or use tools like the SmartObject Service Tester to populate the SmartObject with data.

Adding a property to a SmartObject



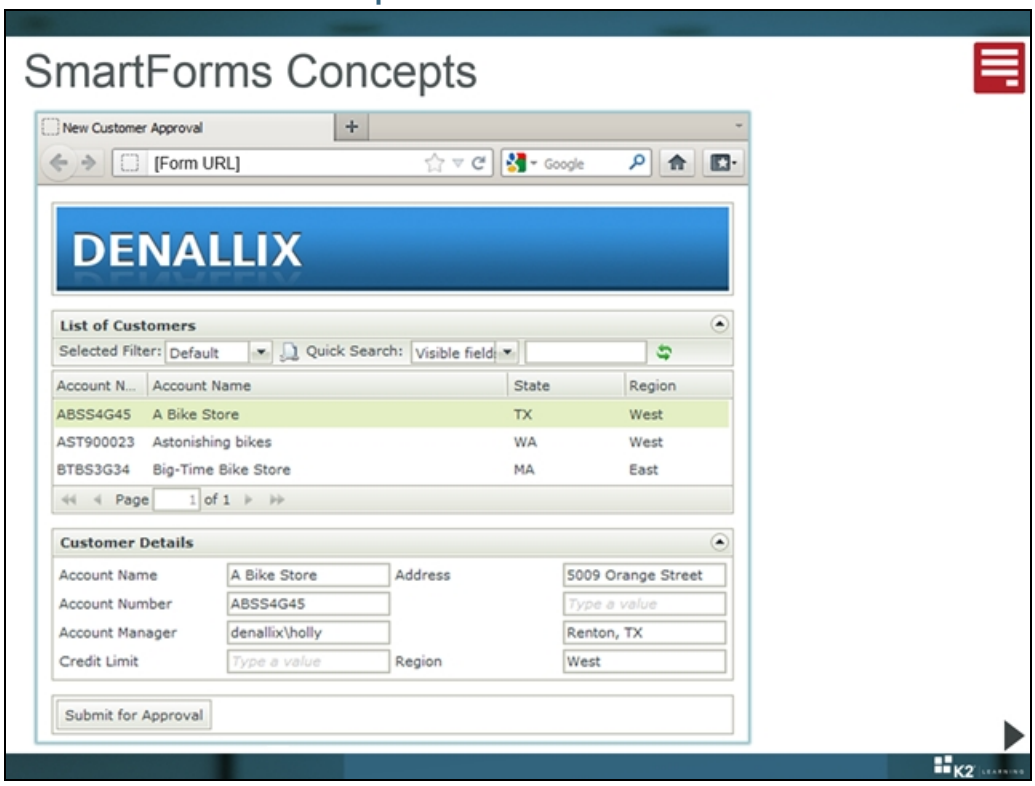
If you want, you can use K2 Workspace to control who may execute what methods against SmartBox-based SmartObjects, or who may create, update or delete SmartBox SmartObject definitions, as shown below:



Summary

- SmartBox is a K2-provided storage area used when you do not already have an underlying system to store the data.
- K2 takes care of creating and maintaining the SQL tables for SmartBox SmartObjects behind the scenes when the SmartObjects are published.
- SmartBox is the default provider for new SmartObjects created in K2 Designer. If you want to use other systems, you have to switch to the advanced mode.
- You can use the K2 Workspace to:
  - Set method-level security for SmartBox (i.e. who may execute which methods).
  - Set security to limit who may publish, update and delete SmartObject definitions.

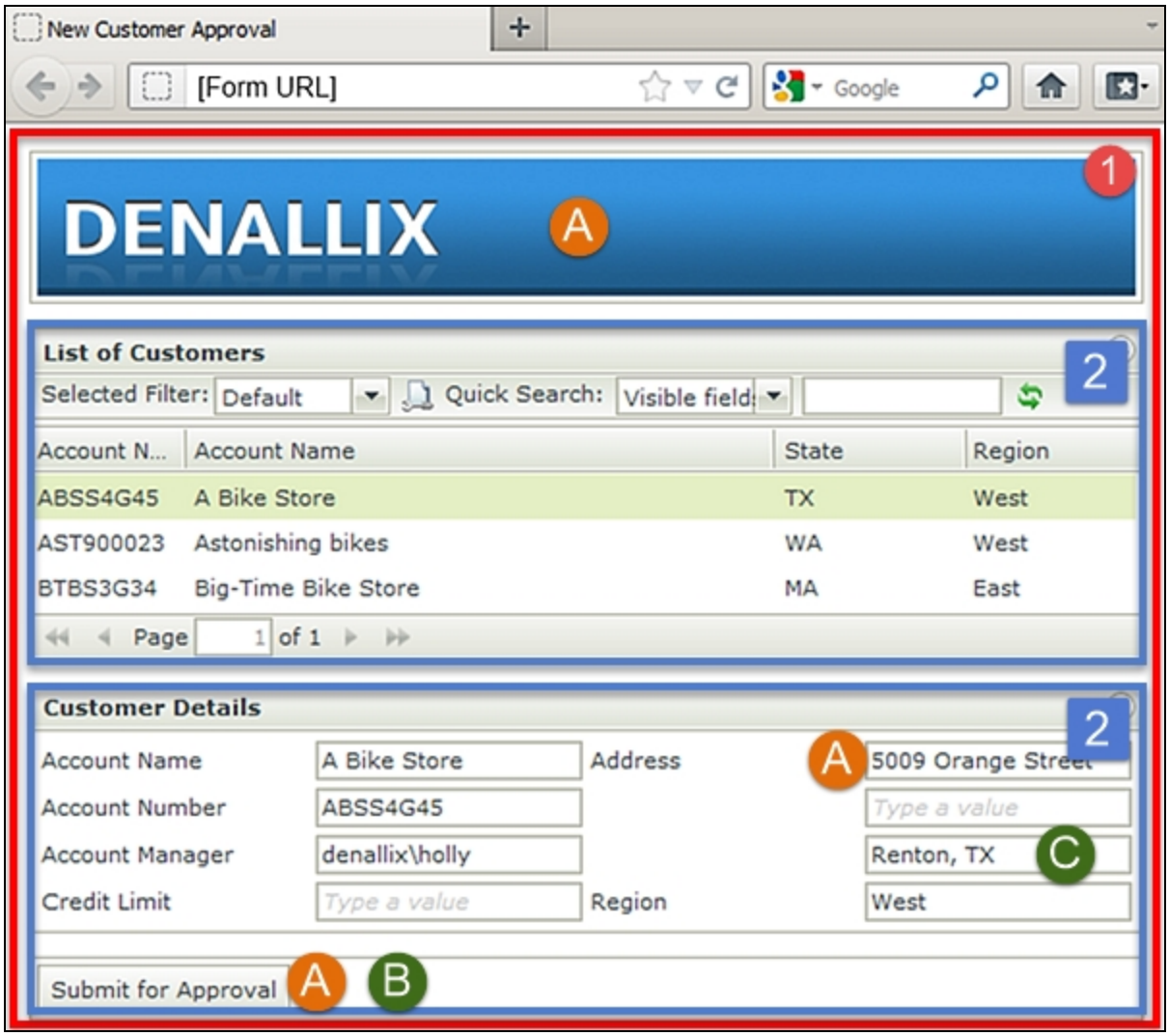
SmartForms Concepts



Video

A SmartForm is made up of several building blocks and it is important to understand how these building blocks are used together to build a SmartForm. In this topic, we will briefly cover the basic components of a typical SmartForm: Form, Views (List and Item Views), Controls, Rules and Expressions. Other topics in this module will dive deeper into these components, but for now we just want you to have an overall understanding how these components work together.

In the image below, we break down the components of a SmartForm for a sample Form. Review the component details that are shown just below the image so that you are familiar with the concepts and terminology of a SmartForm.



1 (Red) **Form**: All of the form contents are contained within the Form.

2 (Blue) **Views**: Forms contains Views. In this sample, there are two Views on the Form: a List View listing several customer records, and an Item View that shows the details for a specific customer. (List Views contains rows of records, while the Item Views contain the content from one record only.)

A (Orange) **Controls**: There is an image control at the top of the form, an Address text box control and a Submit Button control. All of these are different types of Controls that you can add to Views and Forms.

B (Green) **Expressions**: Expressions can be used to format fields. In this sample, Renton,TX is an expression that concatenates the City field and the State field.

C (Green) **Rules**: The Submit Button contains rules that start the associated workflow. Rules might also validate that certain form fields are not empty.

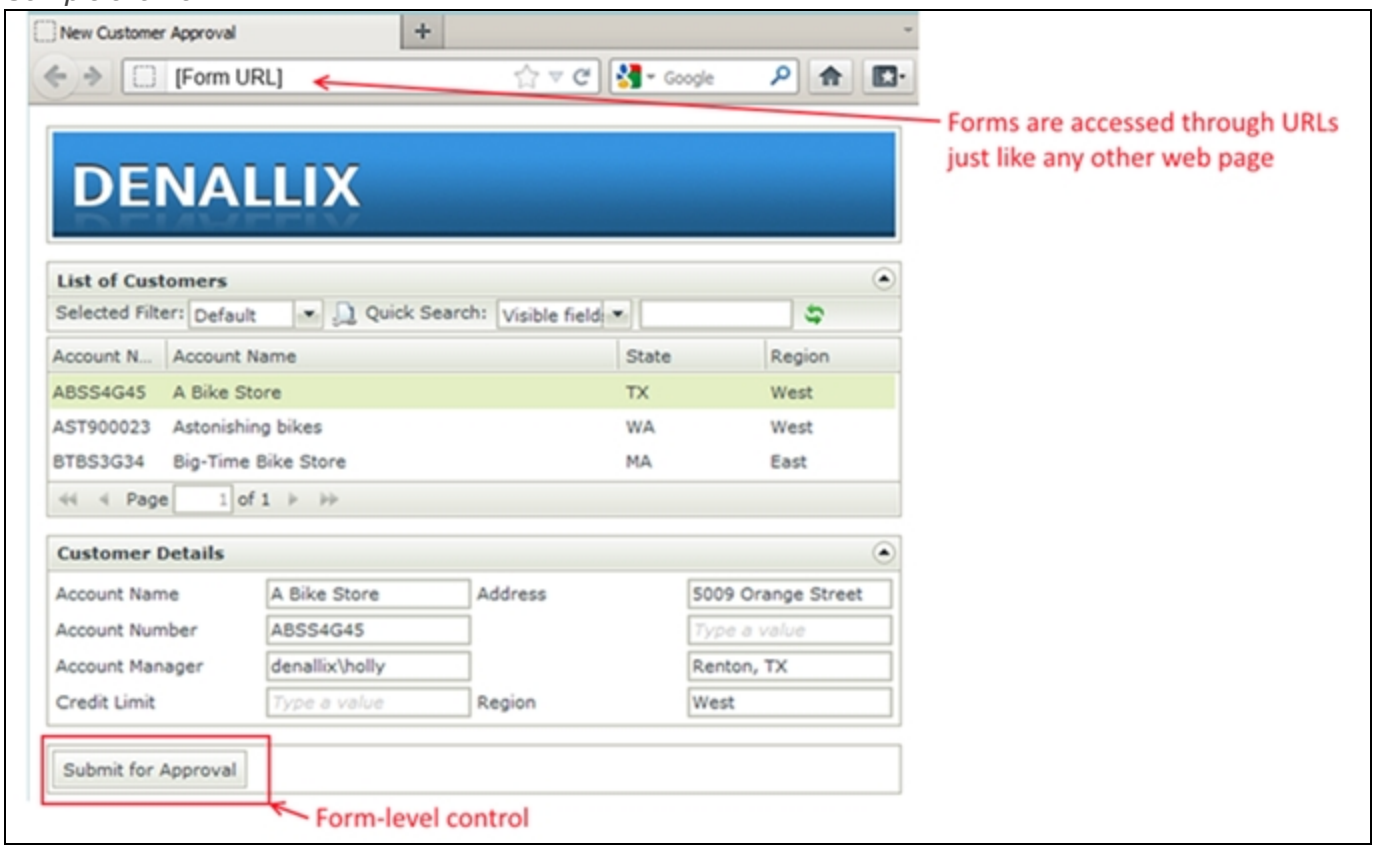
Form

A Form is the web page that is presented to the end user. A Form is a container for one or more Views, and may contain Form-level controls and Form-level rules as well. Note that Forms do not display data from SmartObjects directly: SmartObject data is always displayed in Views. Forms can be “Tabbed” so that multiple views can be presented in the same Form.

Forms can include Form-level controls and rules, and these controls or rules can act on controls and rules in the Views displayed on a Form. Because a Form is aware of the Views that it contains, Form-level rules can enable interaction between Views on the Form. For example, a Form-level rule can read values in one View and then pass those values to another View on the same Form. A Form can also call rules on its child Views.

End users cannot use Views directly; they must access Views through Forms. This effectively means that, until a View is added to a Form, end users will not be able to use the View.

Sample of a Form



View

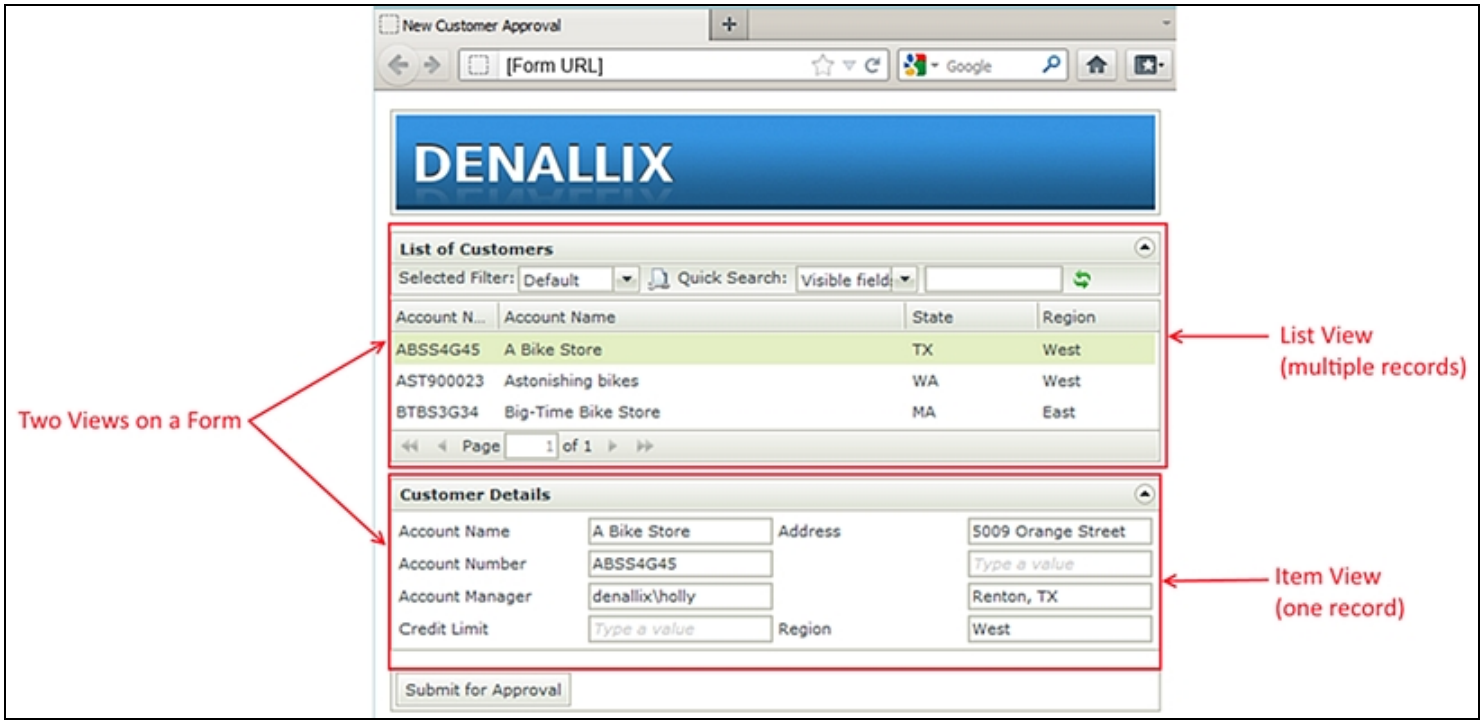
A View visually represents data from a SmartObject and contains controls such as buttons, images, labels or lists of information. These controls may or may not be bound to SmartObject properties. Views are also used to input data, which is then saved into a SmartObject using Rules.

When designing a View, you can select various kinds of controls to represent the properties of the SmartObject, for example a date/time picker control for a Date property, or a File Attachment control for a File property. Views are normally laid out with a table-like structure, and it is possible to modify the table layout by adding columns and rows and merging cells. You can also apply styling to controls so that they are displayed in a specific format in the View, and Views can be expanded and collapsed to control the display of multiple Views on a single Form.

An important point is that a View can be re-used on multiple Forms. This helps to speed up UI development significantly since generic Views can be re-used in many other solutions.

Views can be classified into two main types: List Views (which display data from multiple items) and Item Views (which display data from only one item). A classic example of a List View is a grid-like display where data from multiple records of the same SmartObject is presented in a tabular format. Item Views are used to present data from one SmartObject record. These views are normally used to display the details of a particular record or to allow users to capture/update data for a specific record.

Views on a Form

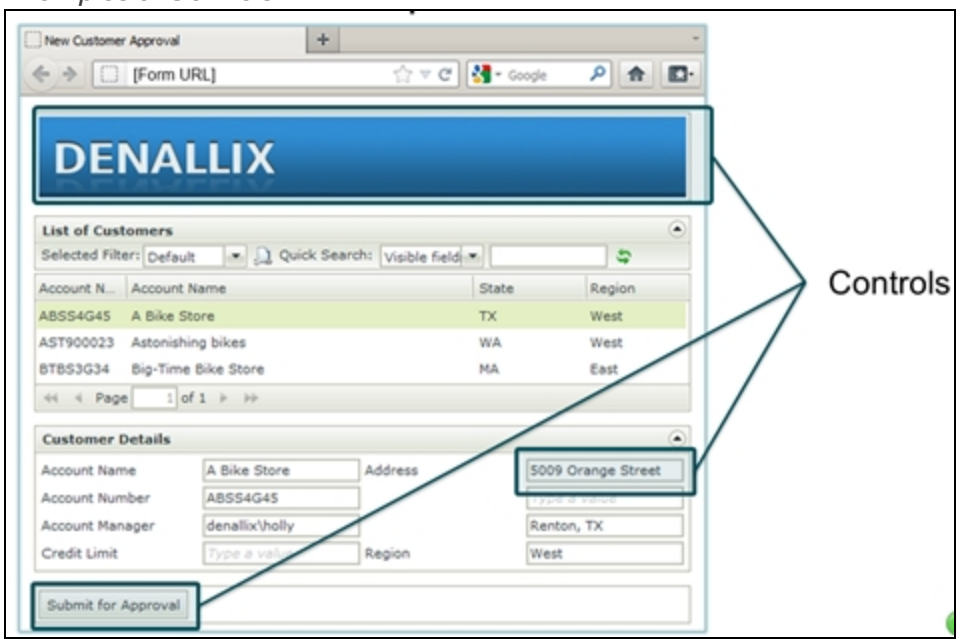


Control

Controls are items like text boxes, labels, drop-down menus, images, buttons and more. Controls can be used on a View or a Form, and are typically used to display the individual properties of a SmartObject, display the result of an expression/calculation, show images or to expose buttons that an end-user can click to execute a Rule.

Controls can be styled (statically or conditionally) using the Style Builder, can have Validation to ensure that user input is accurate, can be populated by Expressions and can have control-level rules that perform some action when an event happens on the control. (For example, as soon as the user changes the content of the Credit Limit textbox, show a message to warn the user that changing the credit limit will require approval.)

Examples of Controls

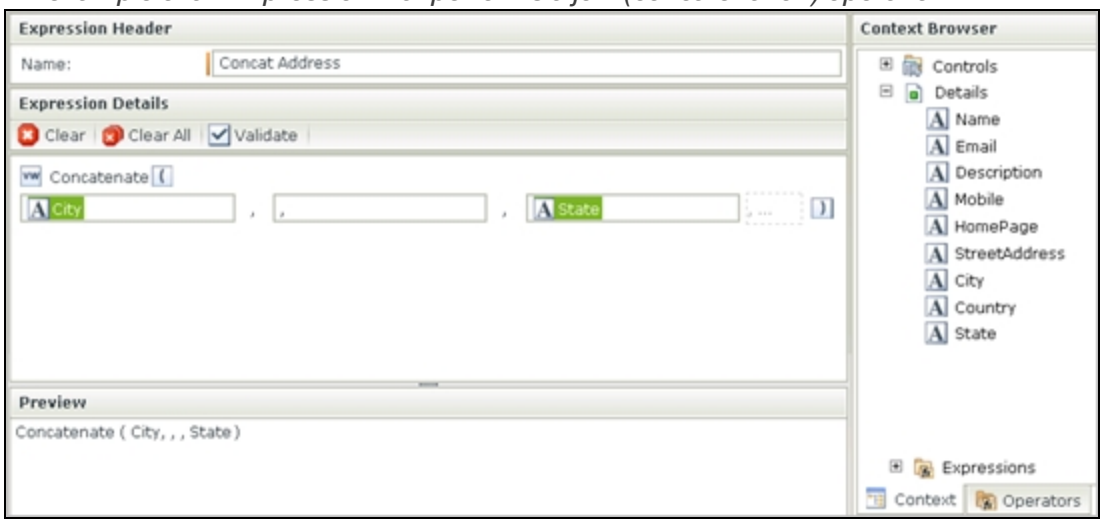


Expression

Expressions are essentially operations or functions that can manipulate data or perform calculations. Normally, Expressions are used to auto-populate controls on an Item View (such as using Text manipulation functions to perform concatenation and search-and-replace-functions or perform mathematical calculations), or aggregate data across Lists (such as calculating a sub-total or average for a list of items).

There is a range of standard Operators that can be used to build up Expressions. The Expression Editor is used to combine these Operators with properties to perform some type of processing. Designers can also use the Expression Editor to define more advanced processing such as nesting expressions and re-using the result of other expressions that exists in the same View.

An example of an Expression that performs a join (concatenation) operation



Rule

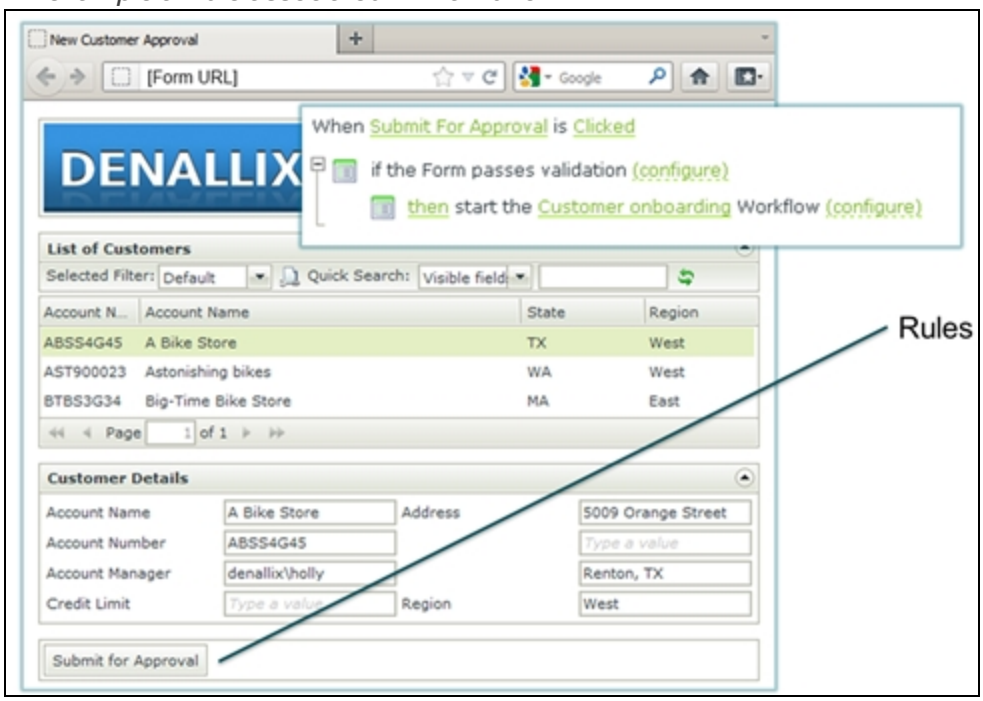
Rules are the business logic behind Forms, Controls and Views, and normally execute an action such as calling a SmartObject method, launching a subform or navigating the user to another page. (There are many more uses for rules, and we will look at these in more detail later in this module and in other modules.)

Rules can be defined for controls (e.g. execute a specific SmartObject method when a button is clicked); Views (e.g. populate the View with information as soon as it is opened) or Forms (e.g. when the Form is opened, use a Parameter passed to the Form, set a value in a View and then execute a SmartObject method).

Rules are made up of three basic components:

1. an optional Event (defines WHEN the rule should fire)
2. an optional Condition (defines conditions that determines IF the rule should execute)
3. an Action (WHAT the rule should do)

An example a Rule associated with a Button

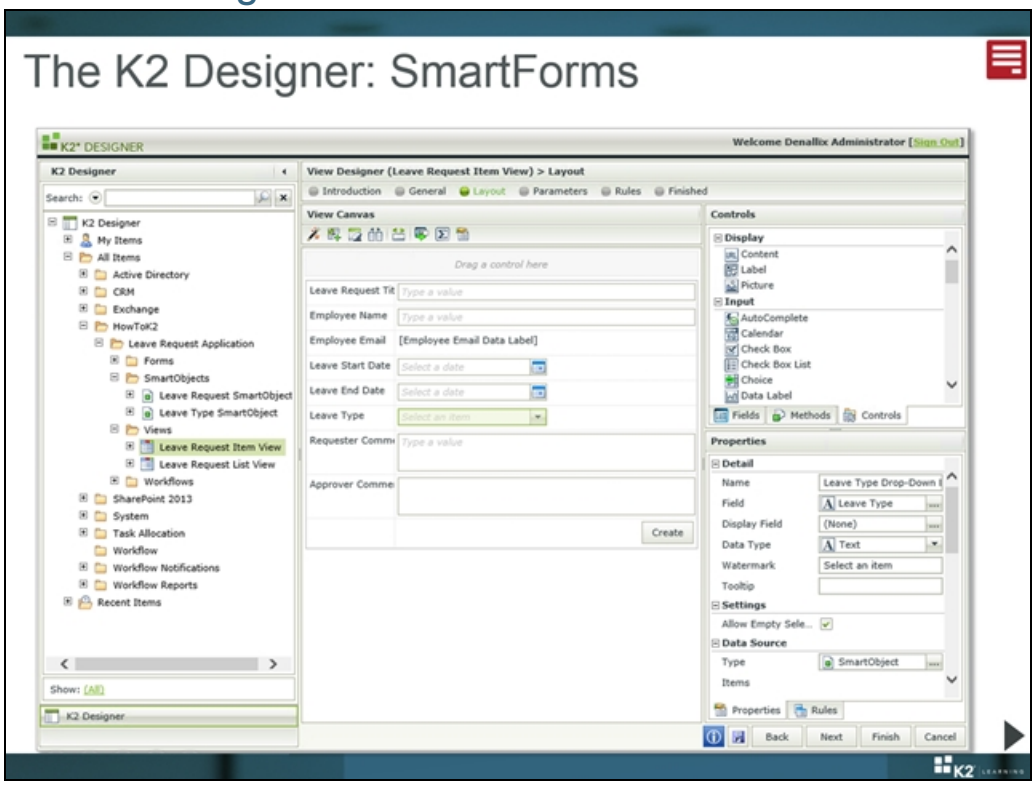


Summary

The basic components of SmartForms are:

- Forms
  - User interact with Forms through URLs as if they were web pages
  - Forms contain one or more Views
  - Forms could have Controls and Rules as well
- Views
  - Views are used to display data from SmartObjects
  - Views are hosted in Forms
- Controls
  - Control are used to display or input data (such as textboxes) or to issue commands (such as buttons)
- Rules
  - Rules are the programming logic for SmartForms and consist of Events, Conditions and Actions
- Expressions
  - Expressions are essentially calculations that are performed against values on a Form or View

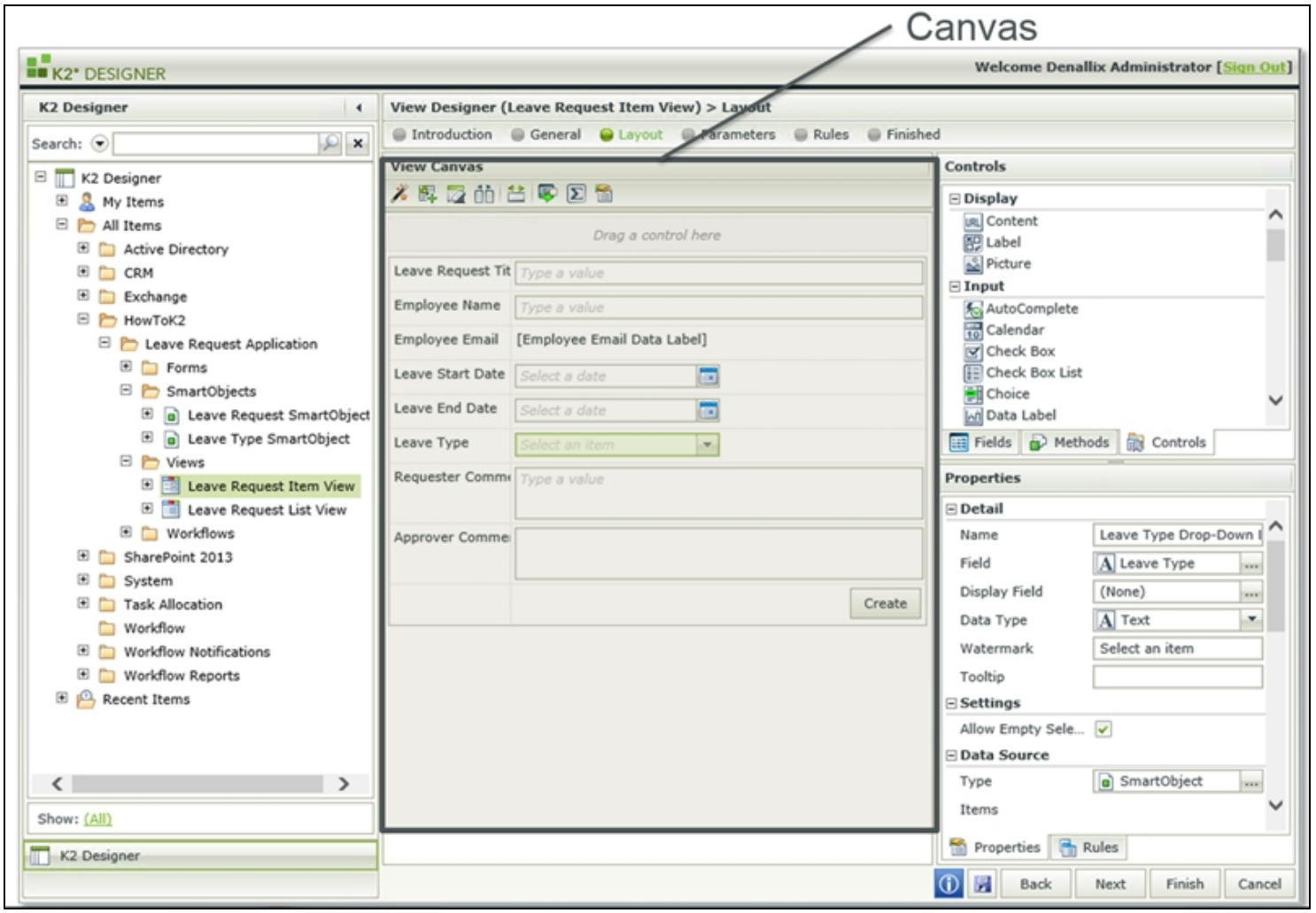
The K2 Designer: SmartForms



The K2 Designer is the only tool you can use to design SmartForms. Let's take a few minutes to get familiar with the Design canvas when building SmartForms.

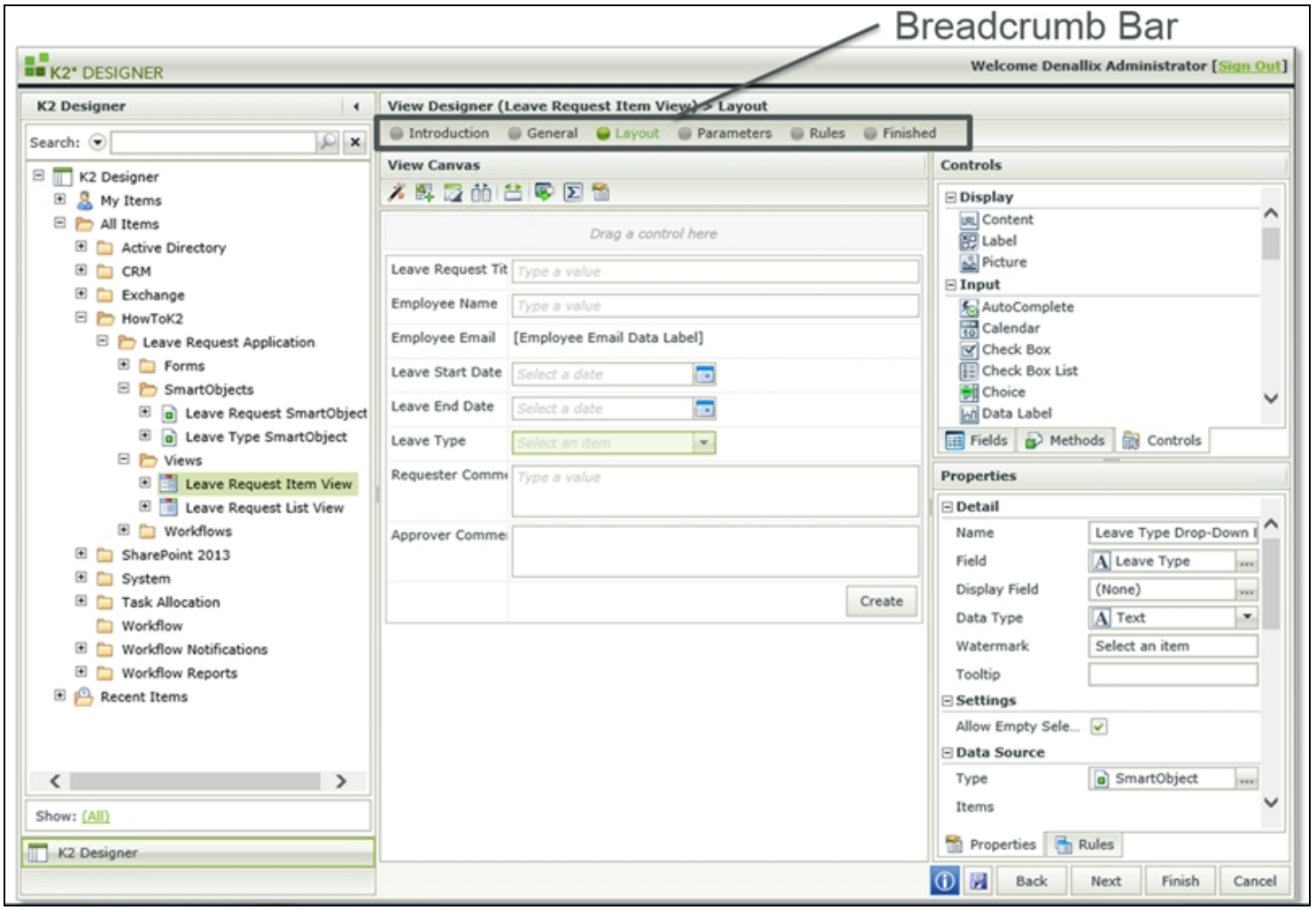
The Canvas is the design area where a Form or View layout is designed. Think of this as the “layout pane” or “editing pane”. When designing a SmartForm, this where you can drag and drop controls and tables into.

The Canvas



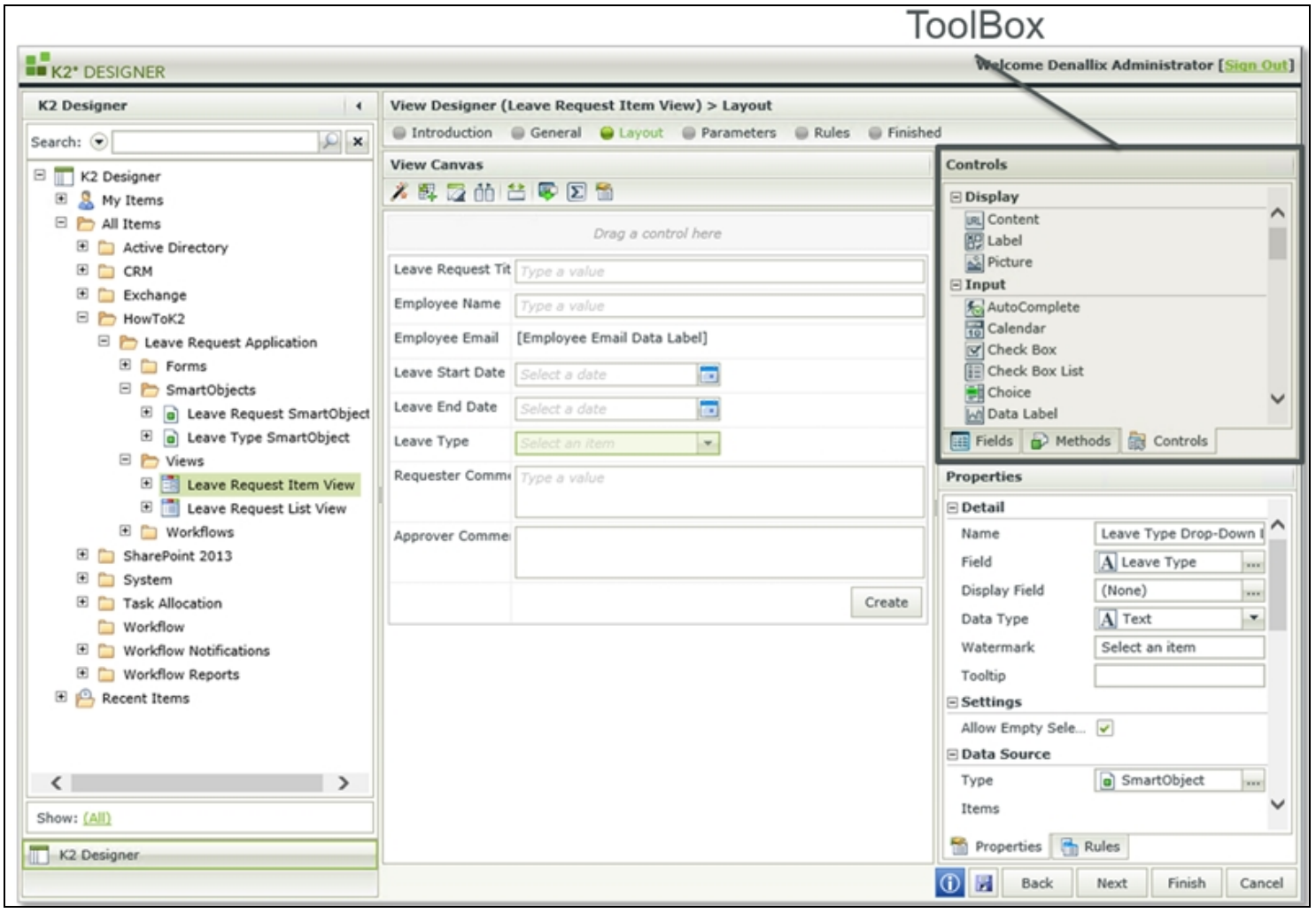
Just above the Canvas, you will find the Breadcrumb Bar. This is a very useful feature, which allows you to jump quickly between various configuration screens of the View or Form, such as defining the Settings, Layout and Rules for the View or Form. Remember to use the Breadcrumb Bar: it will make your life easier when designing SmartForms.

The breadcrumb bar



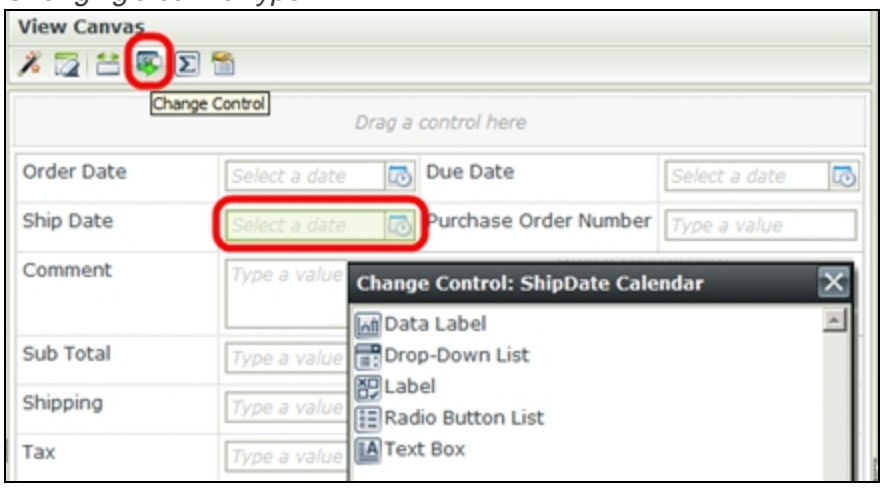
The Toolbox is where you will find the SmartObjects Fields, Methods and Controls that can be added to the design canvas. Think of the Toolbox as a repository of the things you can potentially add to the View or Form. Notice that the Toolbox has three Tabs: Fields, Methods and Controls

The Toolbox



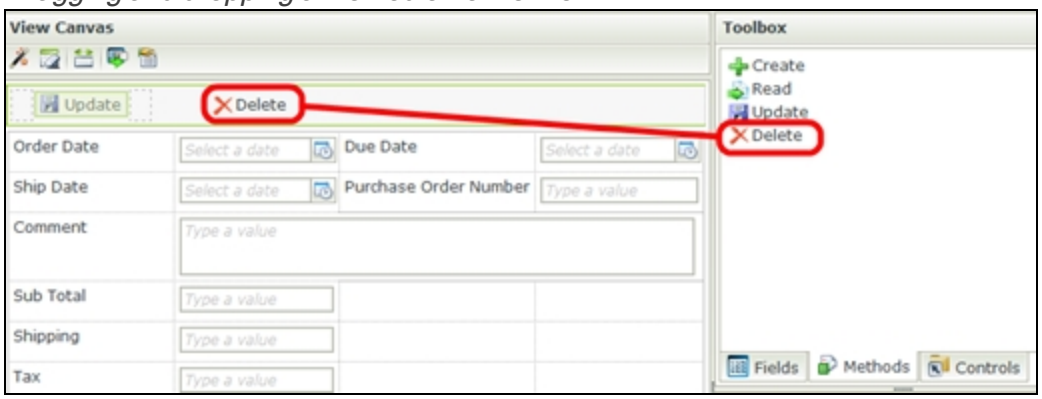
Fields are the properties of the current SmartObject. You can drag these properties onto the design canvas and K2 will automatically generate an appropriate control for the type of field (for example a Date picker control for a DateTime property). If required, you can use the **Change Control** menu button to change the control into some other type.

Changing a control type



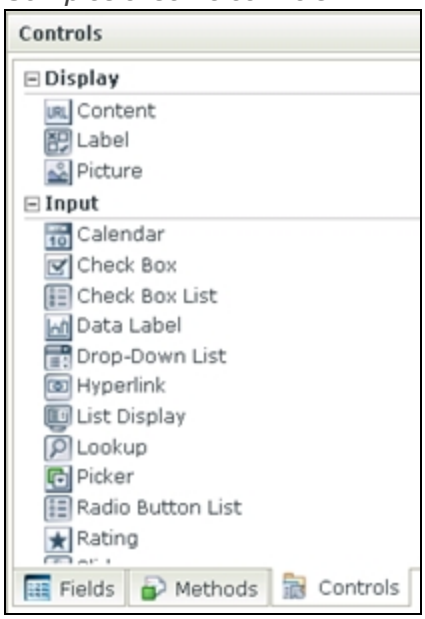
Methods contain the methods that have been defined for the SmartObject. You can drag and drop methods into the View or onto the View Toolbar to add a control that is automatically wired up to execute that method of the SmartObject when it is clicked.

Dragging and dropping a method onto the View



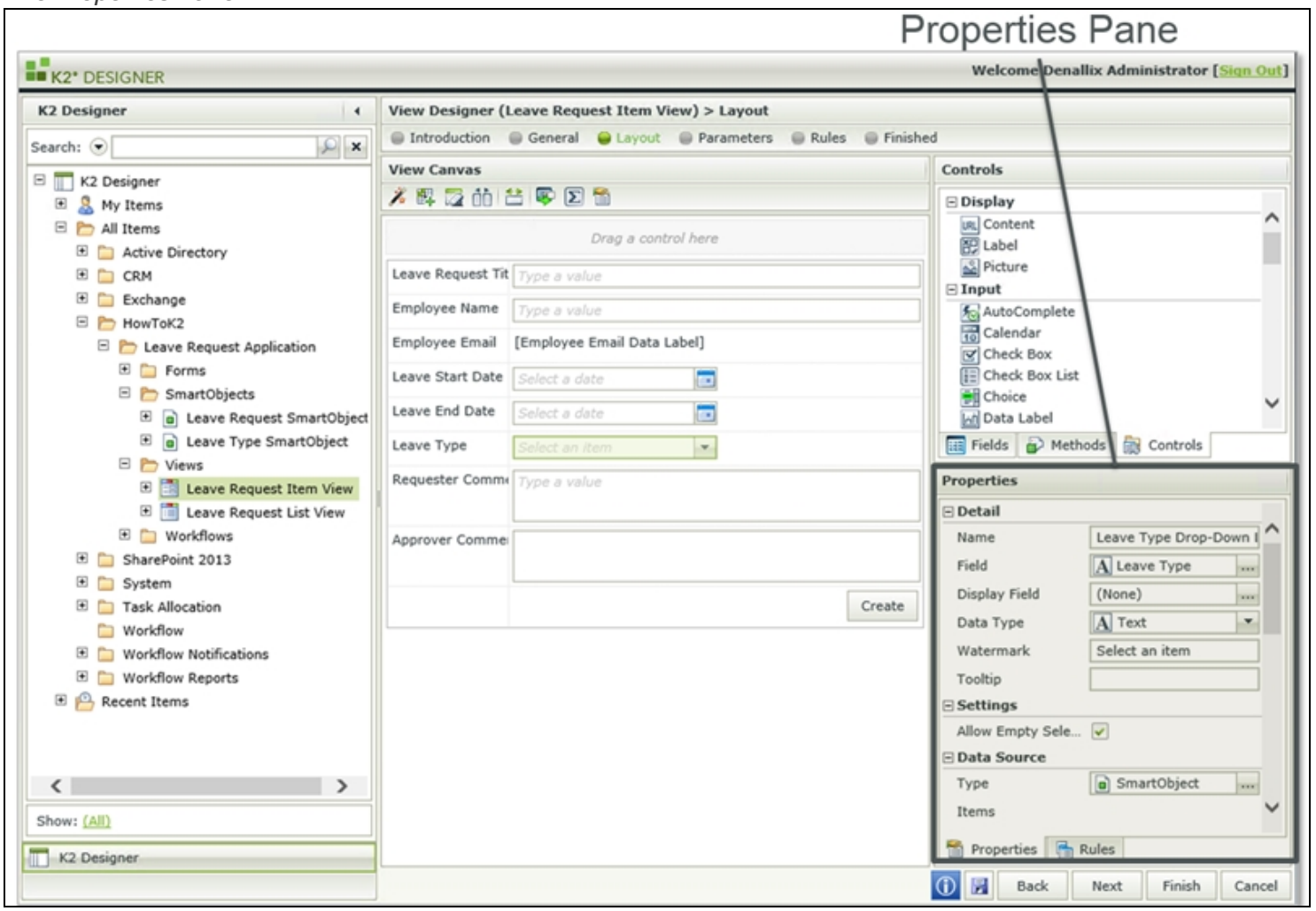
Controls contains a selection of controls like text boxes, labels, drop-down menus, file attachments, picture boxes and more that can be dragged and dropped onto the design canvas. You would normally configure the controls after dropping them on the canvas with the options found in the Properties pane.

Samples of some controls



When you have selected a control on the View, you can set the control's properties in the Properties pane. Here you can set the style of the control, which field to display, validation rules and other rules that will fire when something happens with the control selected. The available options in the properties panel will differ depending on the type of control selected.

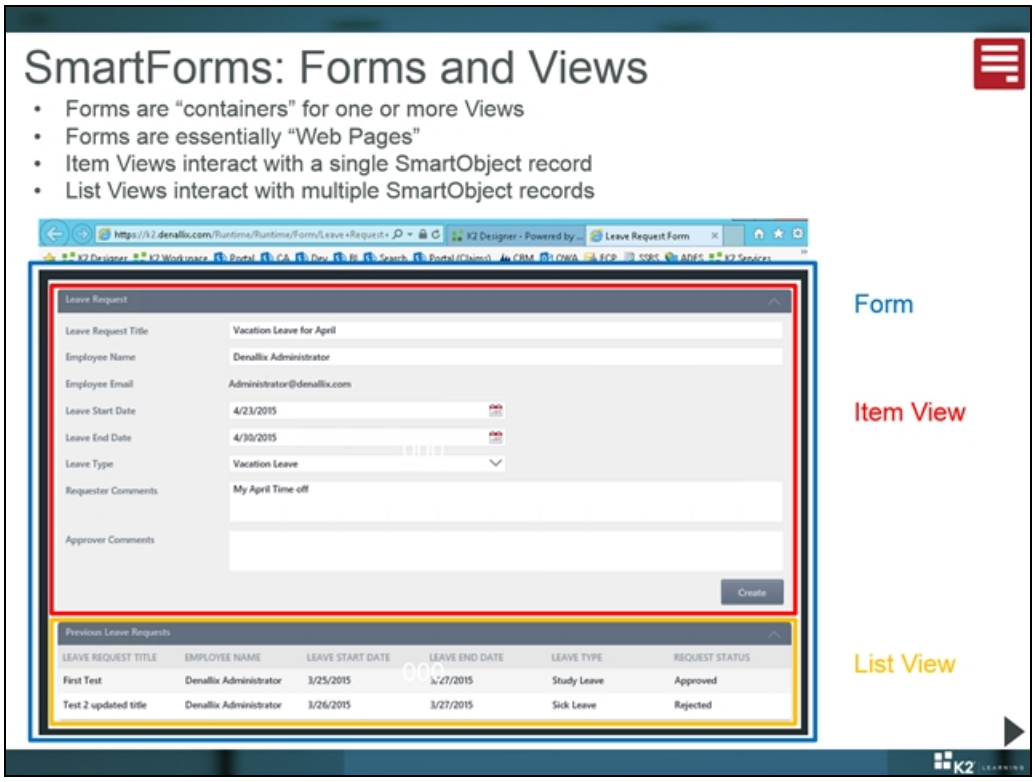
The Properties Pane



Summary

- K2 Designer is where you will design SmartForm Forms and Views
- The designer is separated into different areas
- The Category Browser is essentially a “folder structure” of artifacts
- The Breadcrumb Bar allows you to jump between different configuration screens
- The design Canvas is where you design the Views and Form layouts
- The Toolbox is where you can find Controls, Methods and Field properties from the SmartObject associated with this View
- The Properties Pane is where you configure the item that is selected in the canvas
- Navigation buttons allow you to step through the configuration wizard

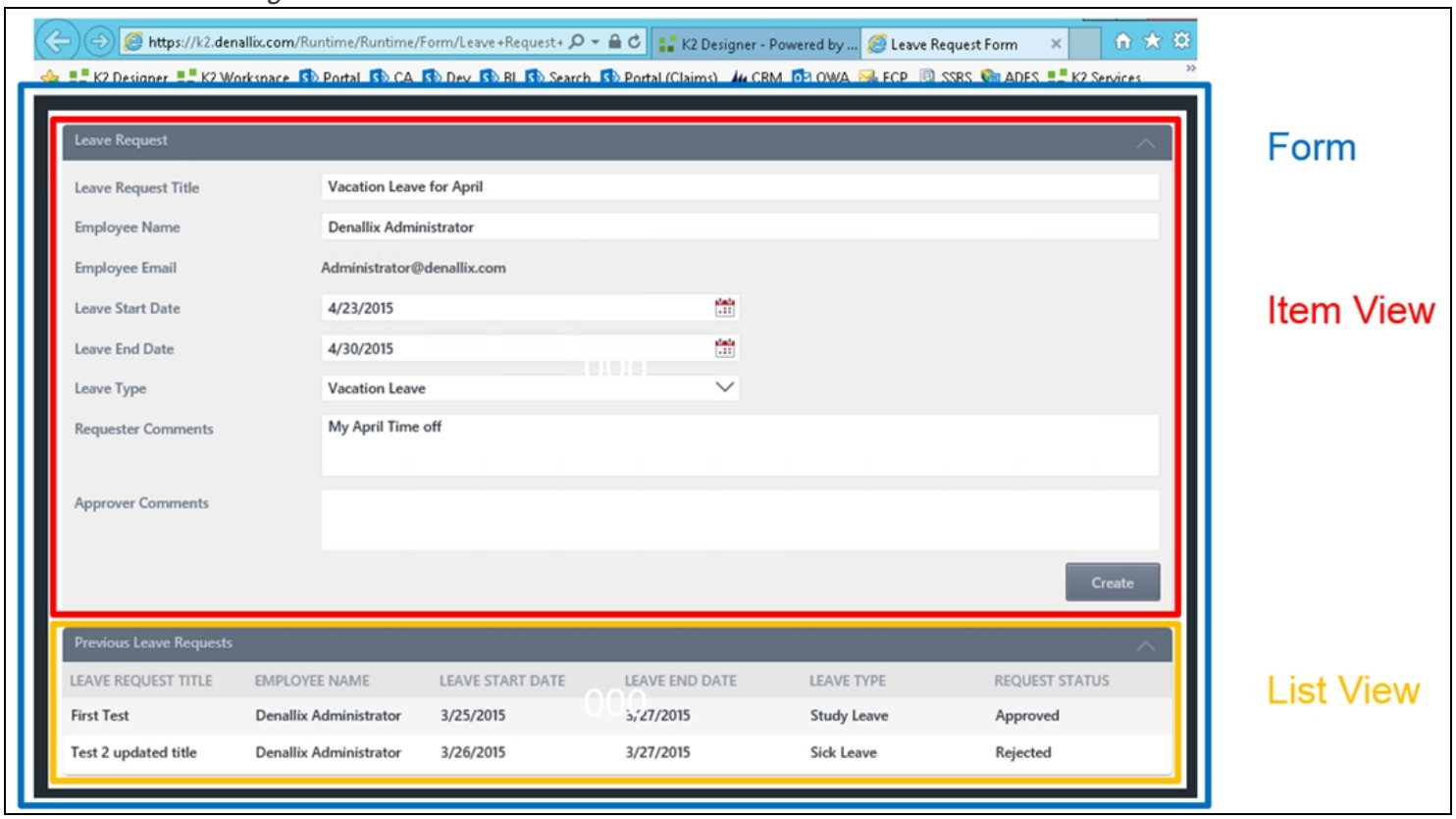
SmartForms: Forms and Views



Video

As mentioned previously, Forms are essentially the "web pages" that end users interact with, and these Forms are exposed as URLs. A Form is a container for one or more Views, and the majority of design and configuration actually lives on the View level. The benefit of this approach is that you can create re-usable Views that can be used on multiple Forms.

A SmartForm showing the Form with two Views: an Item View and a List View



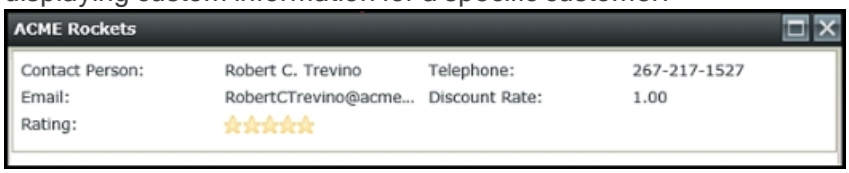
Views are the main components of Forms and as such, you will spend much of your time configuring View layouts and behavior before adding those Views onto a Form. You can add the same View onto different Forms, and this is a nice way of creating re-usable Views and managing those Views in one place. Here is an example: suppose you create a View that displays basic Employee data. You could use this Employee data View on multiple Forms like a Leave Request Form and an Expense Claim Form. If you were to add another Employee data field onto the Employee data View (their Department name, for example), this new field will automatically show up on all the Forms where that View is used, because the Forms only point to the View, they do not actually copy the view.

Views are the building blocks of Forms and are used to display data, capture user input and host other controls such as images, buttons, labels and more. Views are normally associated with at least one SmartObject, and may expose several methods of that SmartObject. Secondary SmartObjects can also be used on the View, for example using a SmartObject to populate a drop-down list on the View. You do not have to associate a View with a SmartObject, however.

As mentioned previously, there are essentially two types of Views: Item Views and List Views:

Item Views

Item views are used to display or interact with a single SmartObject "record" or "entry". These views are normally used in conjunction with scalar SmartObject methods like Create, Update, Delete and Save. In the example below, we are displaying custom information for a specific customer.

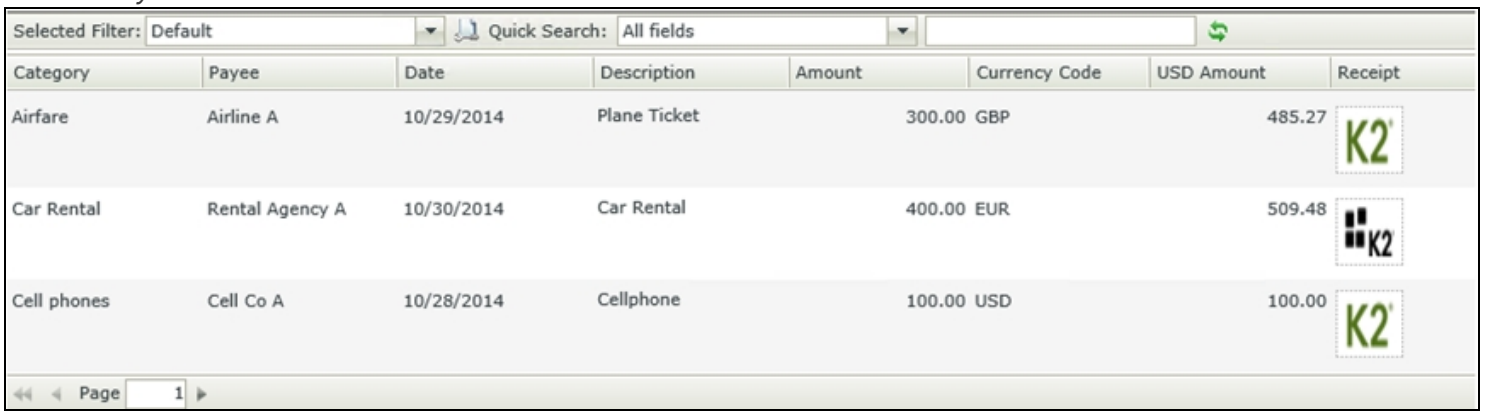


List Views

A List View shows a list of SmartObject "records" or "entries". List Views are used in conjunction with SmartObject List-type methods. Just like Item Views, List Views could be read-only or editable.

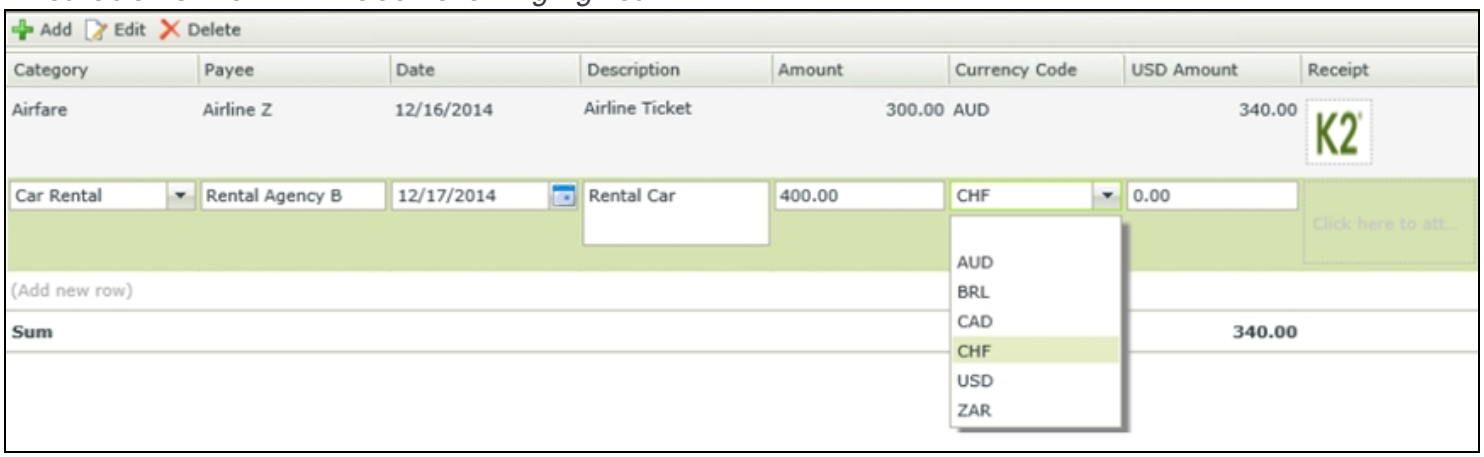
Consider the example List View below, which displays a list of expense claim line items in a grid-like table

A read-only List View



An editable List View is a very powerful feature in SmartForms, and users will love this approach to data entry when adding multiple entries for the same type of SmartObject. Consider the example below: here we are displaying a list of line items for an expense claim. The user can edit the List View by adding and removing items on the list and editing existing items in the list. Behind the scenes, K2 will use the SmartObject's appropriate methods (e.g. Create, Save, Delete) to apply the changes to the SmartObject.

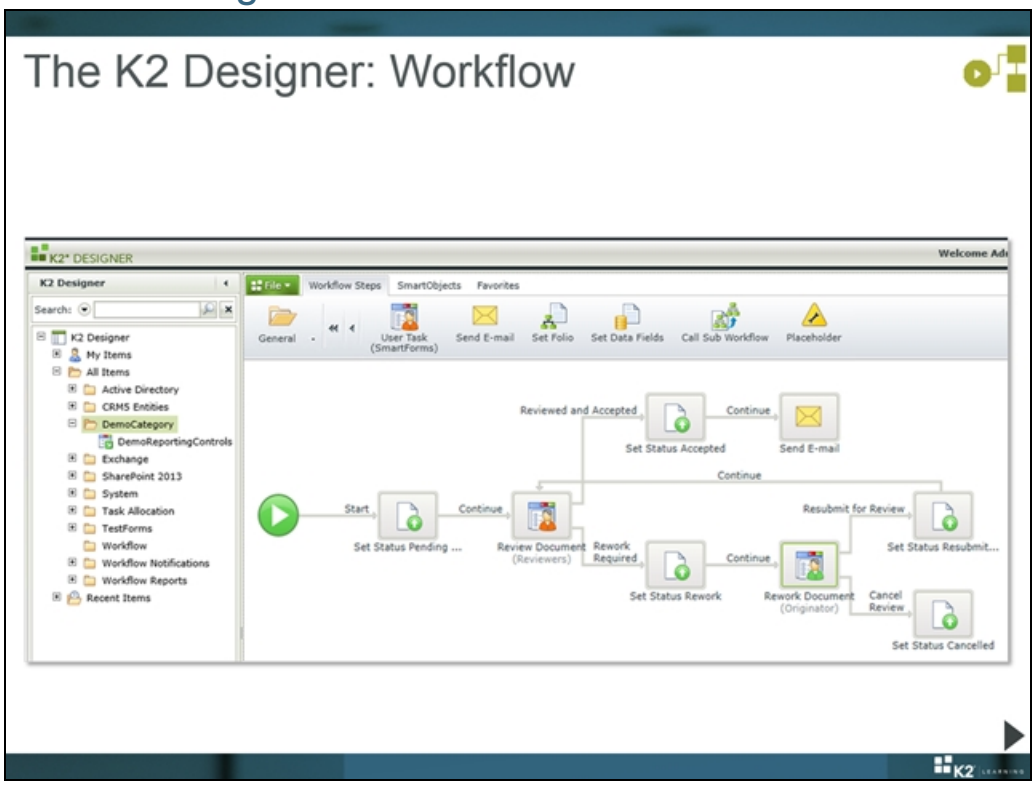
An editable List View with the active row highlighted



Summary

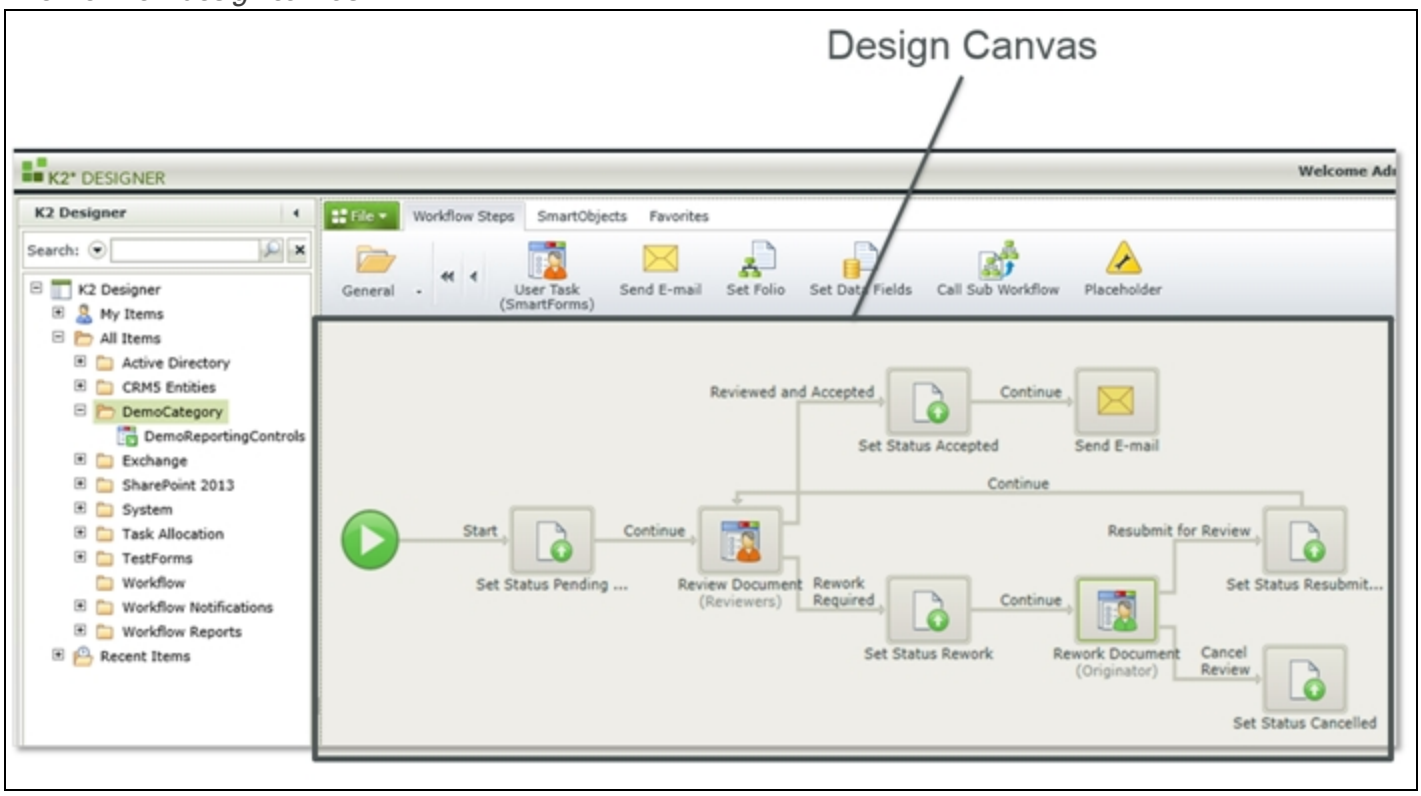
- A Form is a container for one or more Views, and you can re-use the same View on different Forms.
- A Form can contain multiple Views
- Item Views interact with a single record
- List Views interact with multiple records

The K2 Designer: Workflow



You can also build workflows in the K2 Designer tool. The workflow designer looks a little different to the Forms and SmartObjects designers, but the basic operation is very simple: you can add steps to the workflow by dragging and dropping a wizard onto the workflow design canvas as a step, and then configuring the wizard by running through the wizard screens. This workflow design tool does not allow you to drag and drop steps anywhere you want: steps must be dropped into blank steps and K2 will automatically determine the layout of the workflow for you. You can replace existing steps with other steps, and use Outcomes (described later) to add additional paths to the workflow.

The workflow design canvas



The available workflow step wizards are grouped into categories. To select a different type of wizard, click on the category selection button (left-hand side of the steps toolbar) and then select a different category of wizards to show the available steps.

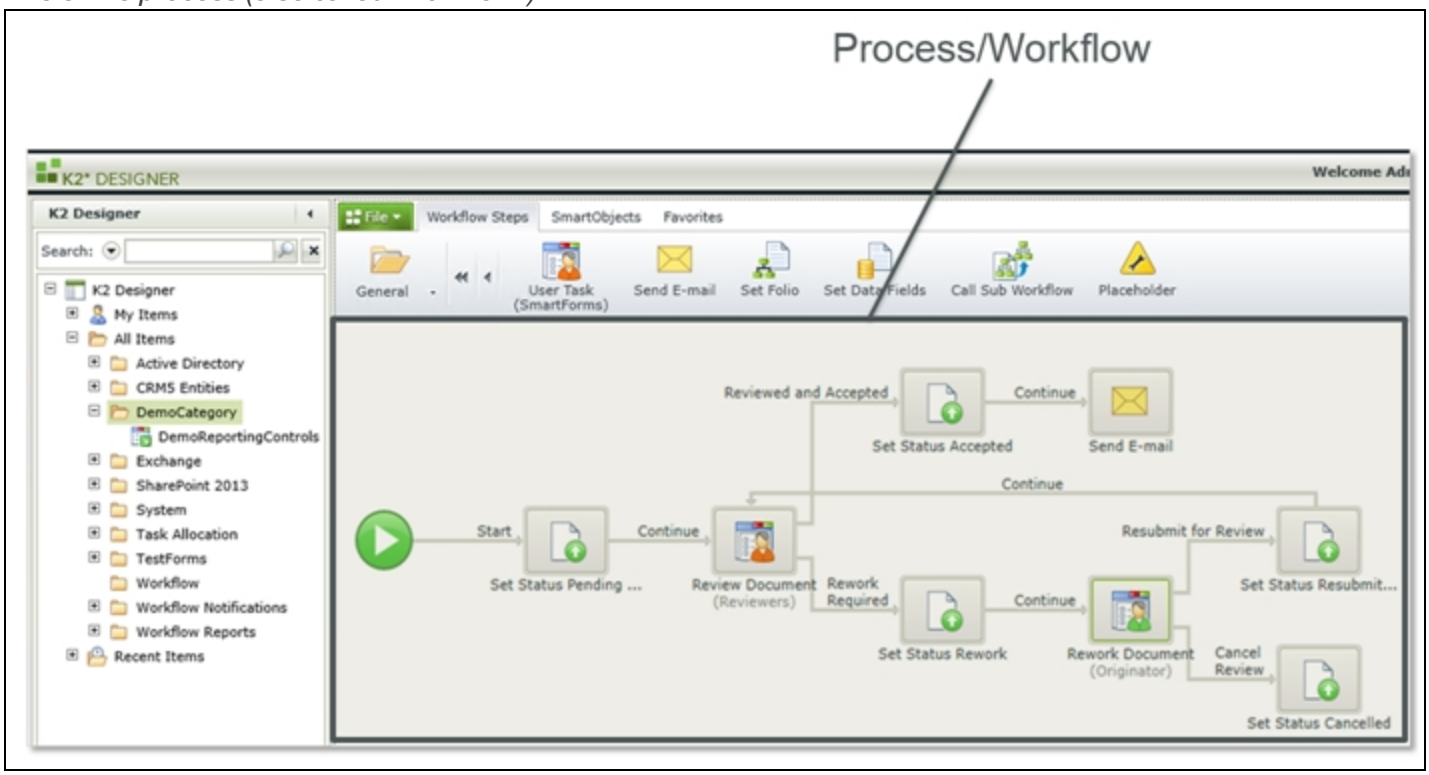
Note
The available wizards will depend on the version of K2 being used and potentially which components were activated in your K2 environment. The screenshot below shows samples of the available wizard types.

The Workflow Steps toolbar

A process (also known as a workflow) refers to the entire process designed in the K2 design tool. A process will have at least a Start step and one Participant or Server step.

Processes usually contain multiple Steps, which are joined with Outcomes. The steps and outcomes define the flow of the process. From a K2 smartforms perspective, SmartForms can be used to start a workflow and can be used as the user interface for User Tasks in a workflow.

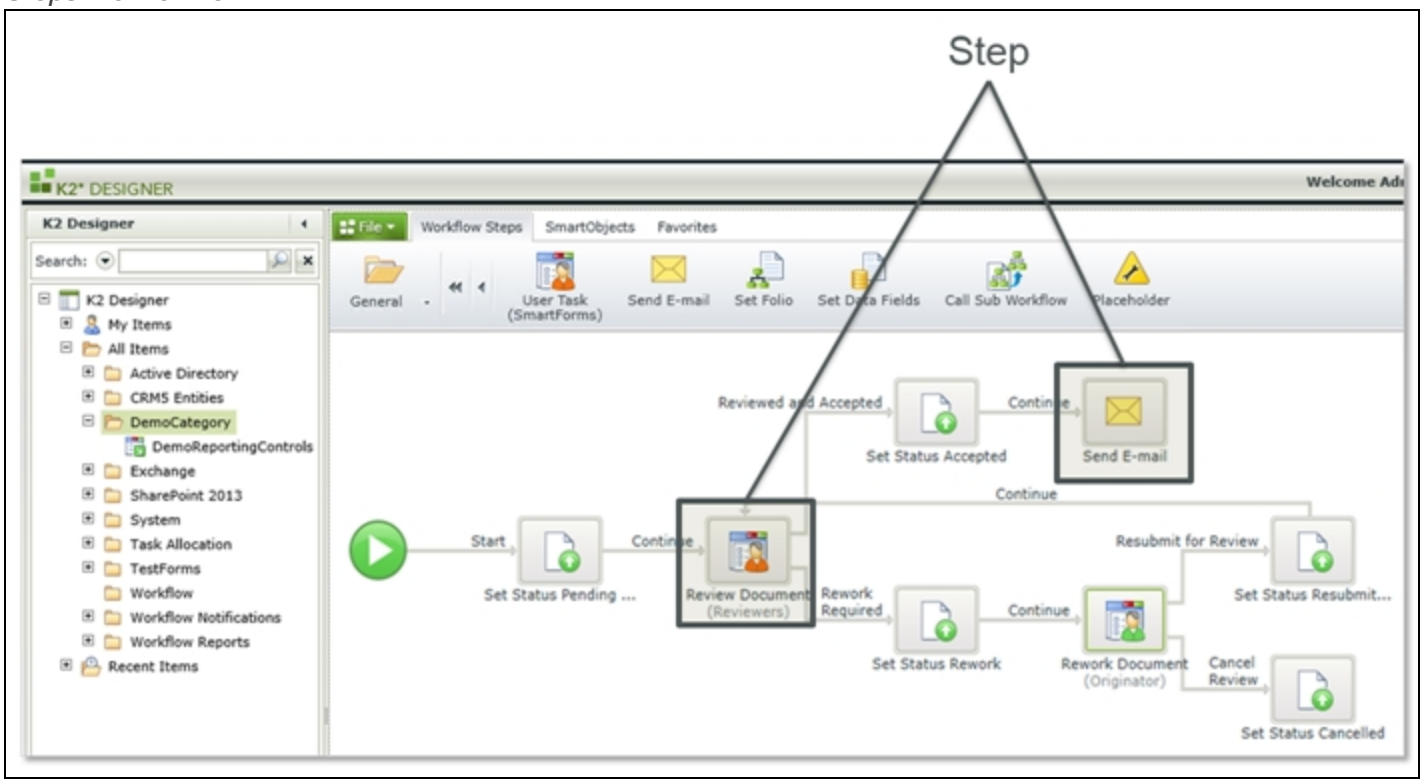
The entire process (also called "workflow")



A Step is a unit of work in a process, which is performed by a human or system. When considering a process design, whenever work moves from one person to another or when the K2 server needs to perform some work, a step will be created for each of these work items.

In the diagram below, two steps are highlighted. In this case, one step (Review Document) is performed by a user, and the other step (Send E-mail) is performed by the K2 server.

Steps in a workflow

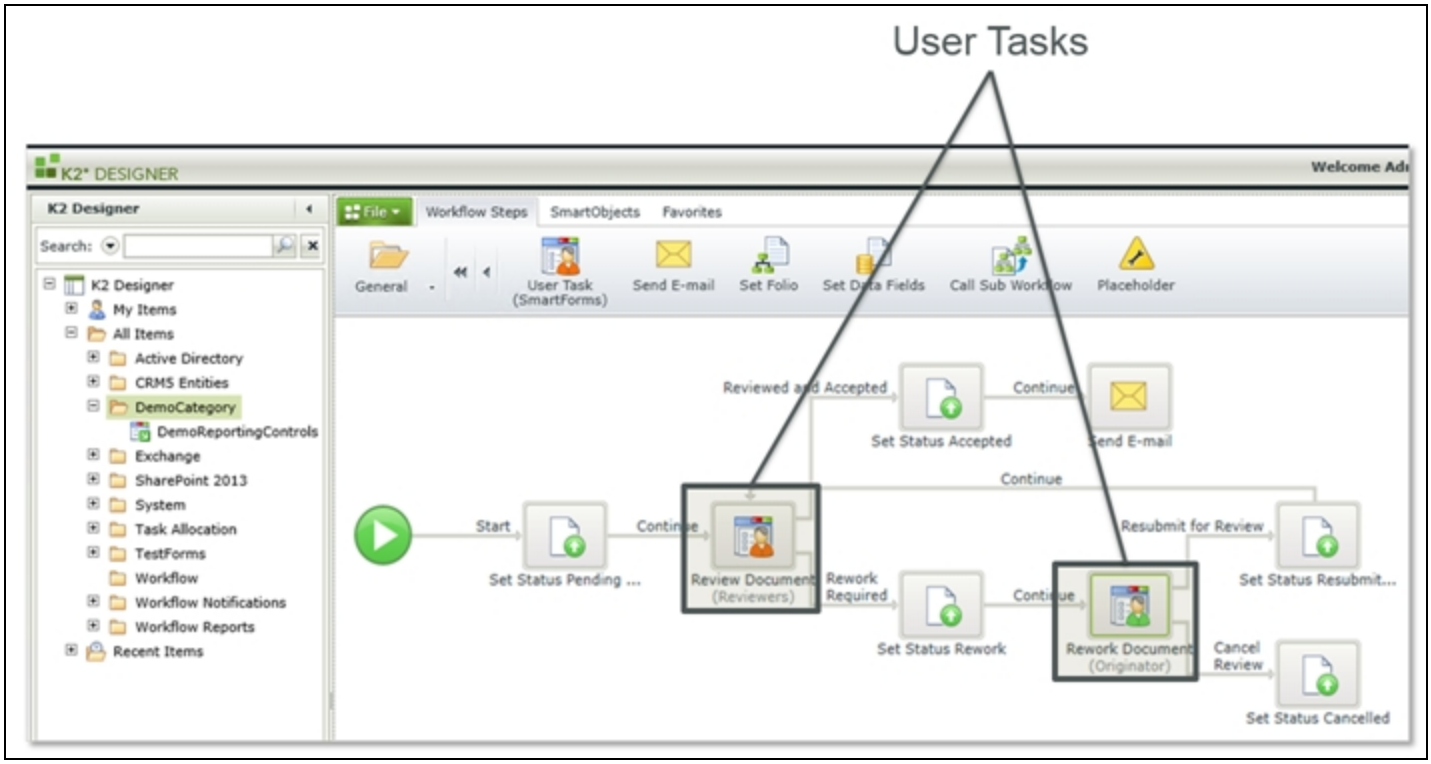


A User Task (also known as a Client Event) is a task that is performed by a human. All tasks that are performed by humans must be allocated to a Participant (also known as a Destination), because a human must perform the task.

In the diagram below, two user tasks have been highlighted. The task with the green user icon is a task that will be performed by the Originator of the process (the person who started the process initially). The task with the red user icon is a task that will be assigned to a Participant, in this case a reviewer for the document.

From a K2 Designer perspective, Forms can be used as the user interfaces for user tasks in a workflow and can be used to start a workflow. You merely need to point the workflow designer to the Form you want to use for a user task, and the designer will do the rest. You can use separate Forms for each step of the workflow, or make use of States and re-use the same Form for multiple steps in the workflow.

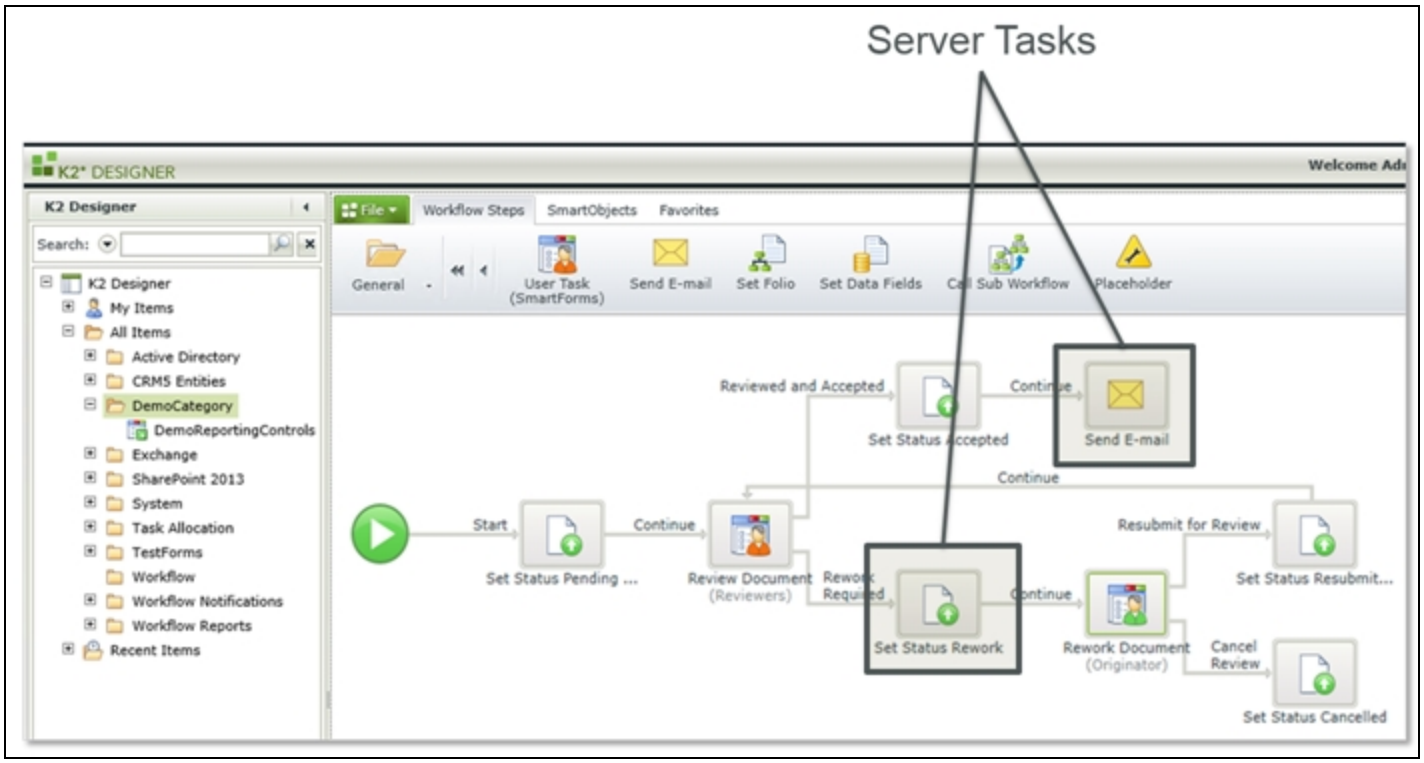
User tasks in a workflow are indicated with user icons



Server tasks are tasks that are performed by the K2 server.

In the diagram below, two server tasks have been highlighted: the task with an envelope icon represents an E-mail that the server will send, while the task with a page-arrow icon represents a task where the K2 server is executing a SmartObject method to update a database with some value.

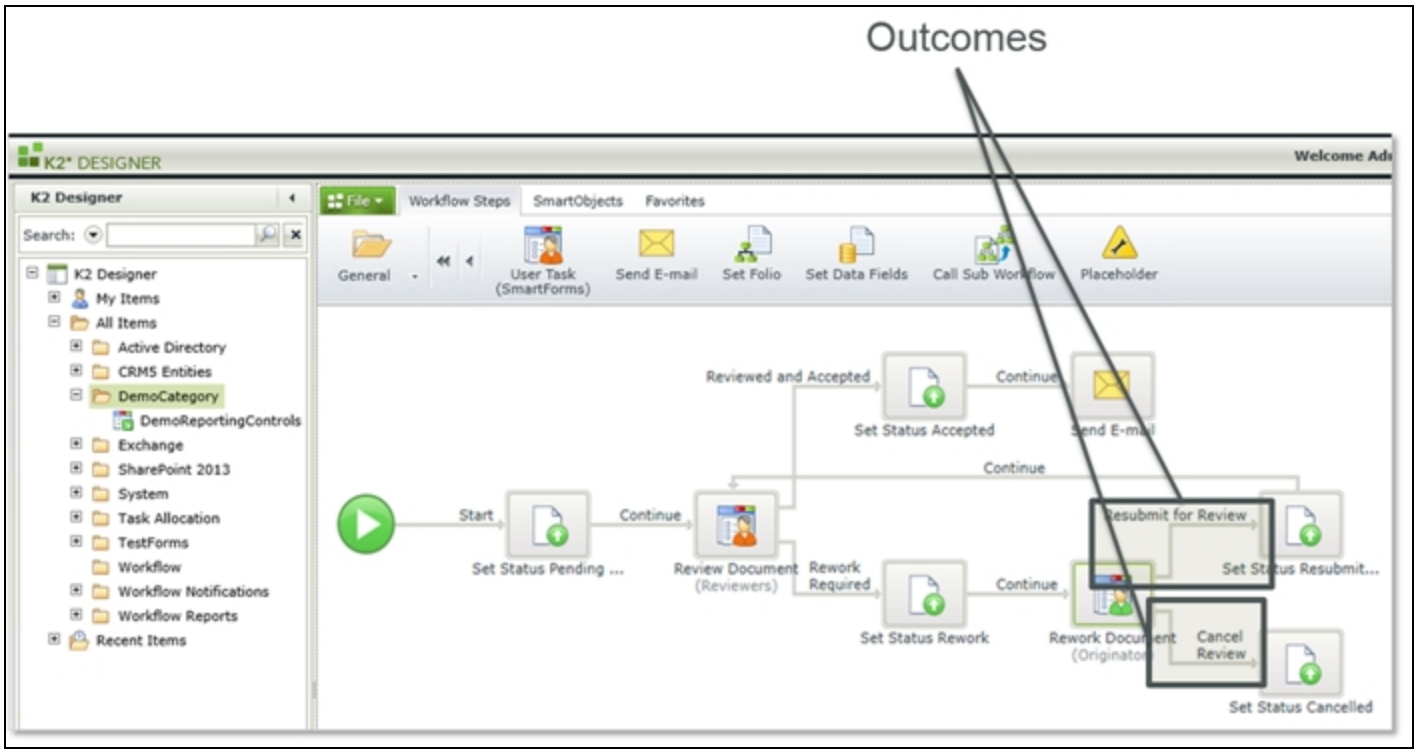
Server tasks in a workflow are all other tasks



Outcomes (also known as Lines) are the possible paths in a workflow. Outcomes may be based on user decisions (Actions), based on some comparison of values performed by the K2 server, or even based on a combination of the two, or just a simple "Continue" outcome where the path is always followed.

In the diagram below, two outcomes from the same task have been highlighted. The “Resubmit for Review” outcome is executed when the originator has finished re-working their request, and the "Cancel Review" outcome is followed when the user decides to cancel their request. While user actions are often associated with Outcomes, an Outcome does not have to depend on a user's decision. A step in the workflow could also have multiple outcomes (for example one for each of the three possible choices a user could make on a task) and could even have parallel outcomes (in other words, the workflow will split into parallel paths that are executed at the same time).

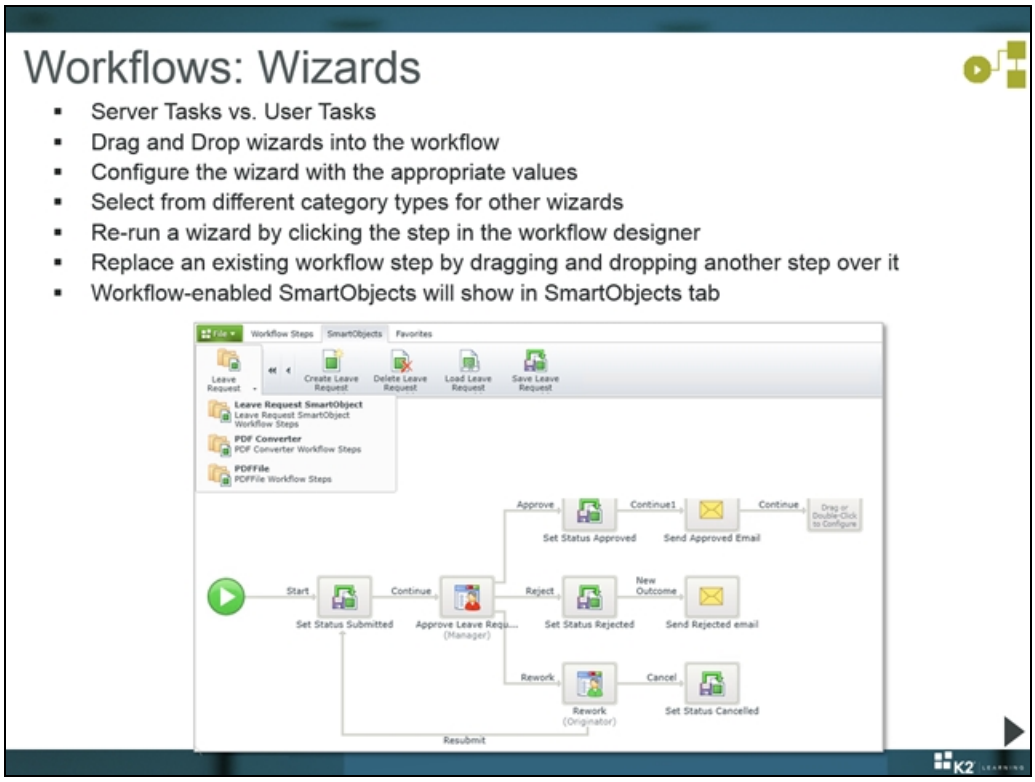
Outcomes are the "lines" in a workflow



Summary

- The design canvas is where you “lay out” your workflow
- The web-based workflow designer does not offer free-form placement of steps (you cannot drag-and-drop steps around in the workflow)
- The step wizards are drag-and-drop “events” or “steps” that you can drop into the workflow and then configure according to the desired functionality
- Click on the category selector to pick a different set of wizards
- A process (also known as a workflow) refers to the entire process and will have at least a Start step and one User Task or Server Step, but usually multiple steps
- Steps are joined with Lines (also called Outcomes) which define the "flow" of the workflow
- Outcomes may or may not be bound to user actions

Workflows: Wizards

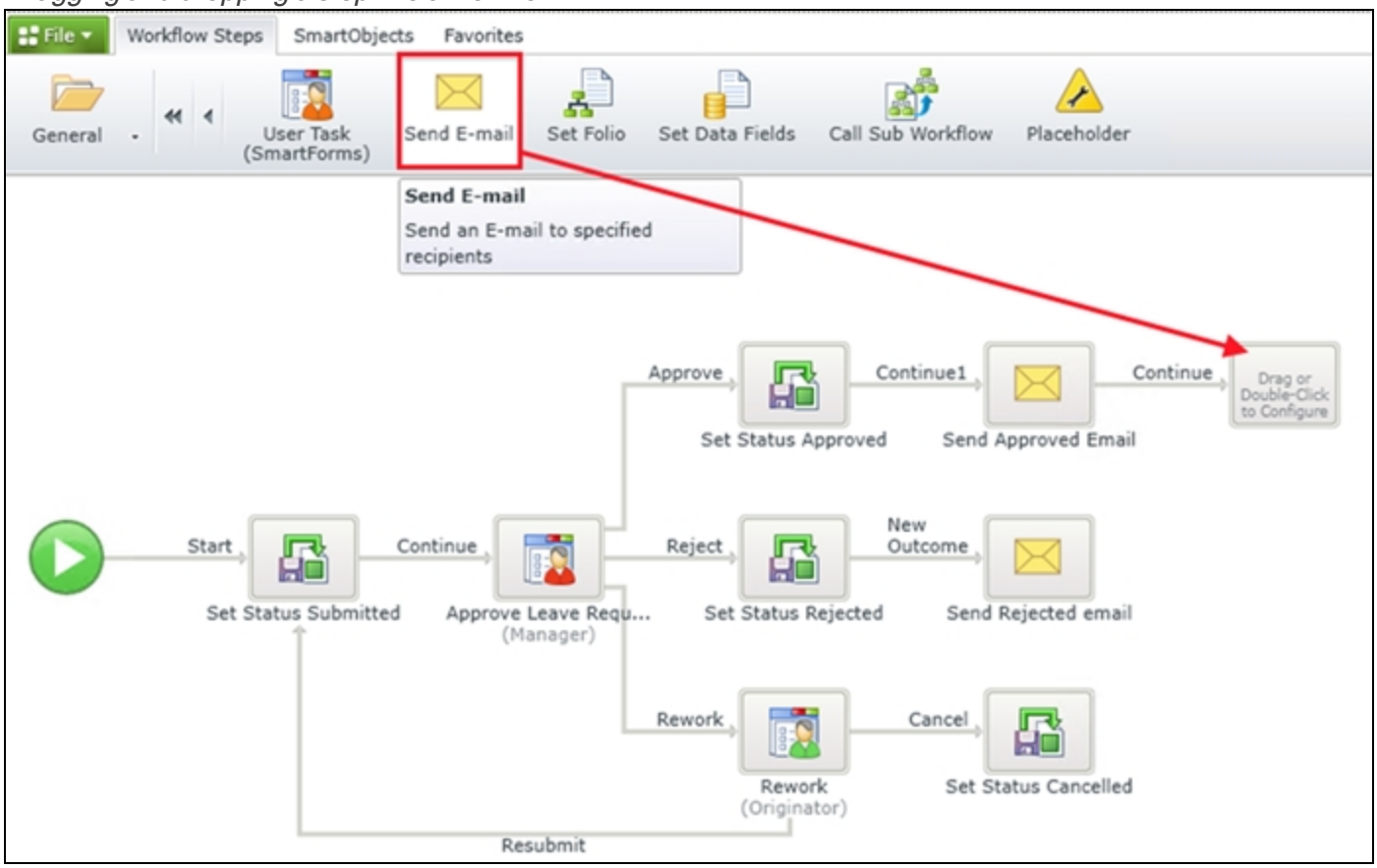


Video

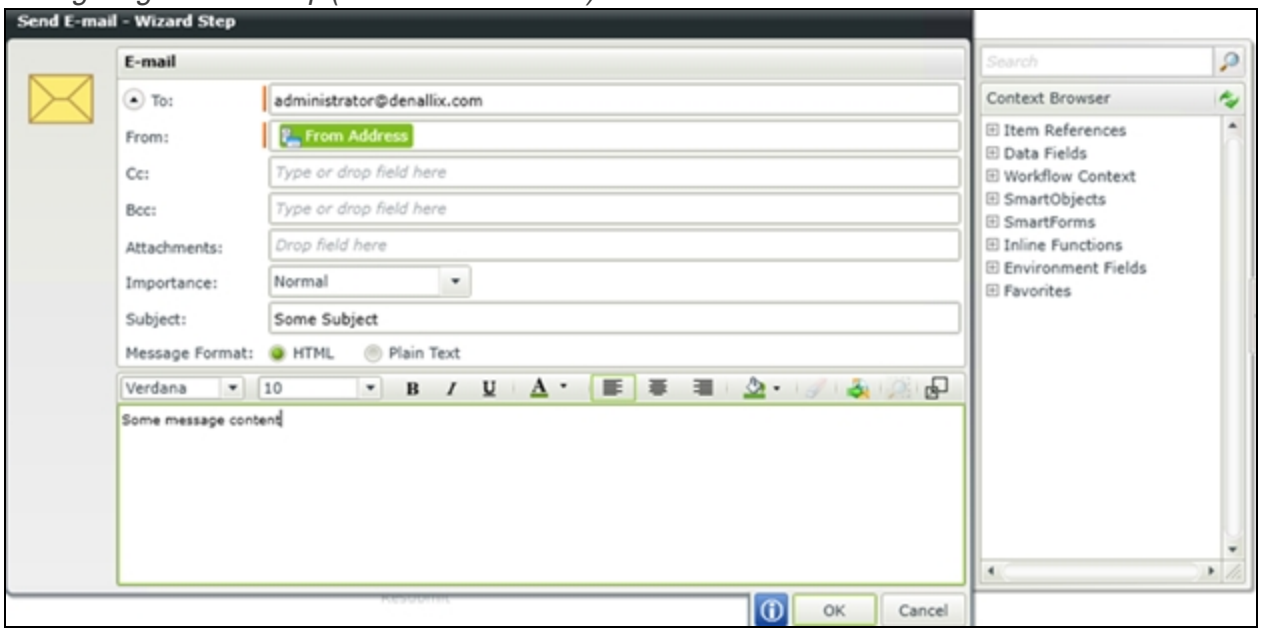
Workflow steps are essentially “tasks” that are assigned to the system (e.g. send an email; these are known as "Server Steps or "Server Events") or actual tasks that are assigned to users (e.g. approve a leave request). The main difference between the two is that once a server task is complete, the workflow will automatically continue on to the next step. However, with a user task, the workflow will assign the task to the task participant(s) and wait until a user completes the task. The workflow will then usually investigate the user's input and, based on the outcome of the step, determine whether it can continue down a particular path of the workflow or if it needs to wait for another user to complete the task.

To build a workflow, you will drag-and-drop a step onto the workflow design canvas, and then run through a configuration wizard to set up the step.

Dragging and dropping a step into a workflow



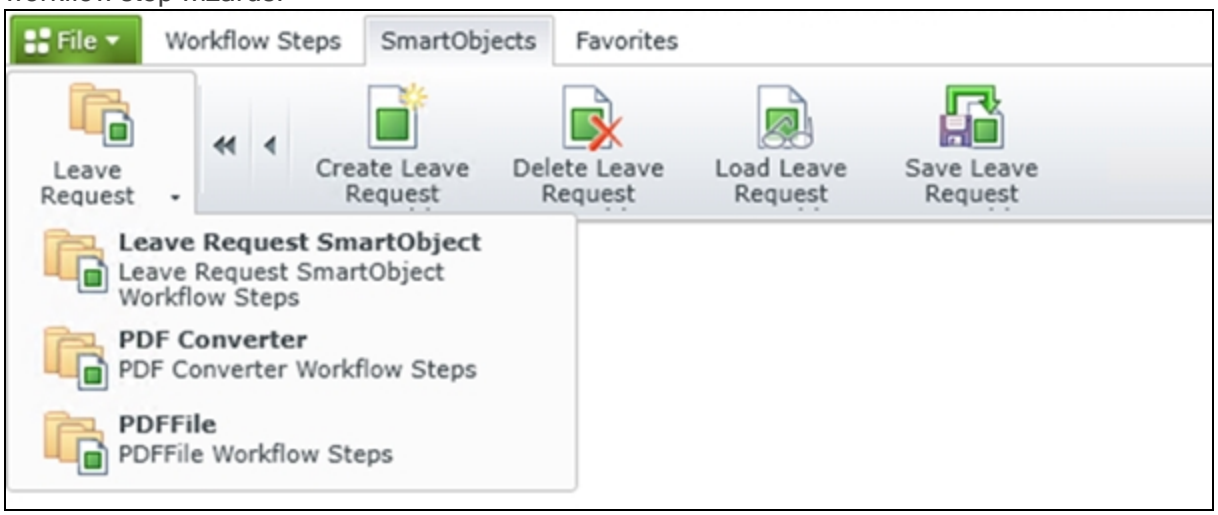
Configuring a wizard step (Send E-mail shown)



To re-run a wizard and reconfigure the values in the wizard, just double-click the step in the workflow designer. You can also replace a step with another type of step by dragging and dropping a step from the ribbon bar into an existing step of the workflow.

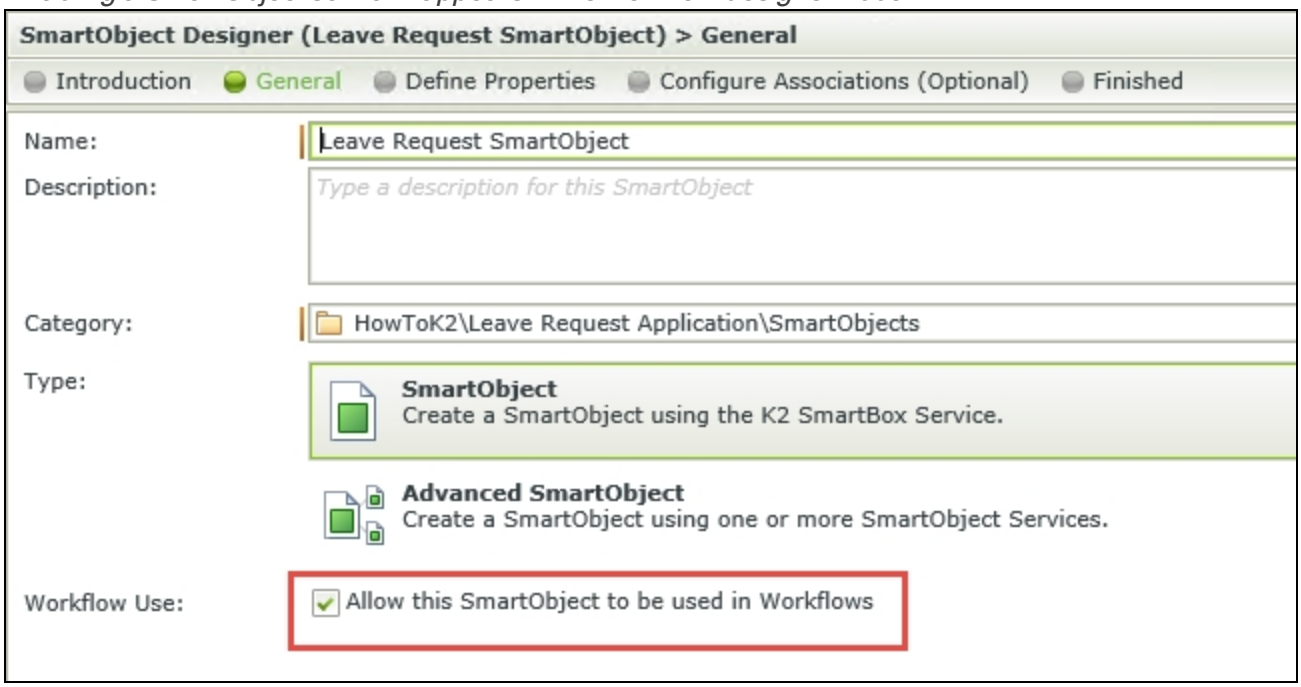
If you have workflow-enabled SmartObjects, you can find those SmartObjects in the **SmartObjects** tab of the workflow designer. From here, you can drag-and-drop the method you want to call into the workflow just like any of the standard

workflow step wizards.



To expose a SmartObject on the workflow tab, select the **Allow this SmartObject to be used in Workflows** option, which is shown on the first screen (General) of the SmartObject designer wizard, in the K2 Designer.

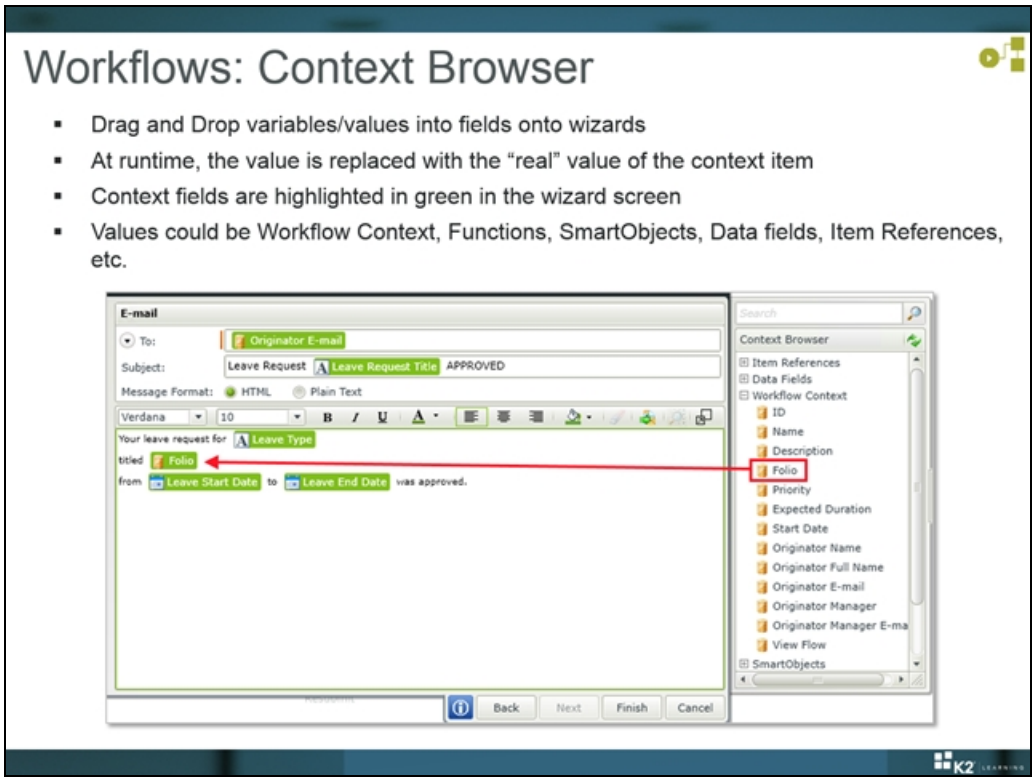
Enabling a SmartObject so that it appears in the workflow designer ribbon



Summary

- Server Tasks are tasks that the K2 Server will perform, such as “send email” or updating a SmartObject. The workflow continues immediately after the server task is done.
- User Tasks are performed by people (specific users or groups) and the workflow will wait for the user’s input before continuing on with the workflow.
- You assemble workflows by dragging and dropping “wizards” into the workflow.
- Configure the wizards by providing values (hardcoded or variable values) for the wizard fields.
- You can re-run a wizard by double-clicking the step.
- You can replace an existing step just by dragging and dropping another workflow step over it.
- Use the placeholder step for a workflow step that does nothing.
- Select different wizards by choosing another category.
- Workflow-enabled SmartObjects will show up in the SmartObjects tab, you can drag and drop these methods into the workflow.

Workflows: Context Browser

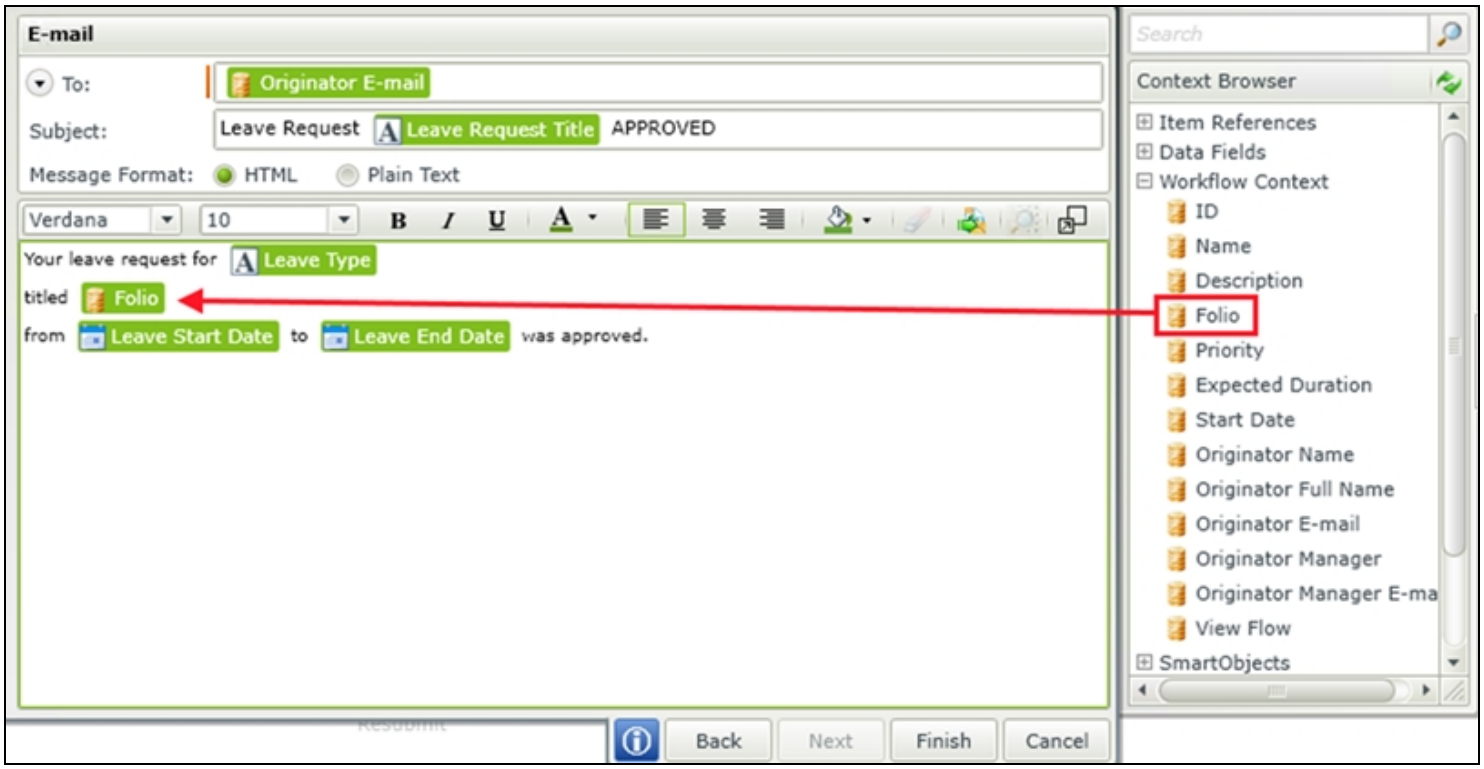


Video

Many workflow wizards provide a Context Browser located on the right-hand side of the window. The Context Browser contains properties and values that you can drag and drop into the workflow wizards. The K2 Context Browser plays a vital role during the workflow design phase because it allows process designers to select variables when designing a process. At runtime, the K2 server will replace those variables with the "real" values when the process is executed.

Consider the example below. Here, the E-Mail event wizard is being configured, and the workflow designer has dragged values from the Context Browser into the E-Mail wizard. The context fields are highlighted in green, indicating that they are variables that K2 will replace at runtime.

The Context Browser window



The real power of the context browser lies in the different categories of context values that are available. Let's look at the available categories.

Category	Details
Item References	This category exposes the Item References (think of these as primary SmartObject records) that relate to the process, like the request data or a document. Item references allow you to easily drop values from this record into workflow wizards.
Data Fields	The Data Fields category exposes any data fields or XML fields defined in the workflow. These Data Fields are usually only relevant to the current workflow instance.
SmartObjects	This category exposes any SmartObjects that have been marked as being available for use in the workflow. If you had a SmartObject that exposes some external system, this is where you would find it.
SmartForms	This tab is available when your Workflow is associated with a SmartForm, and exposes the Views and Controls on the Form.
Inline Functions	K2 provides many functions that can be used in wizard text boxes as well as SmartObject methods to perform processing on data before or after it is used. The available functions are divided into different types depending on the function being performed, such as Date/Time processing, Mathematical functions, Data Conversion functions or Text functions.
Environment Fields	This category exposes server-level configuration settings, both standard settings (such as the SharePoint server URL) as well as custom settings. The ability to add custom settings is especially important when you want to deploy solutions across different environments. Here is an example: suppose that each of the environments at the organization (Development, Test and Production) have a different address that should be used when sending E-Mails as part of a workflow. If users receive an E-Mail from "testworkflow@denallix.com", they

	know that this is an E-Mail form the test environment. When the workflow is deployed to Development, Test or Production, it would be tedious to replace the “From” email address for each E-Mail event in the process before it is deployed. A better approach would be to define a custom field for the Environment called “FromEmailAddress”. The value of this field would be different for each environment, and the workflow designer just has to drag the “FromEmailAddress” context field into the E-Mail wizard. When the workflow is executing in Development, K2 will use the development email, and when the workflow is executing in Production, K2 would use the production email. Fields from the Environment Library can be used in K2 workflows as well as K2 smartforms.
Favorites	This category exposes any Inline Functions that you may have saved as a Favorite, allowing you quick access to commonly-used expressions.
Workflow Context	The Workflow Context category exposes various properties of the current workflow, for example the workflow folio, start date, the originator of the workflow and several of the originator’s properties as well as context-sensitive items such as the current activity’s start date. This category is very often used to obtain the originator of the workflow so that a task or email can be sent to that person.

Summary

- You can drag and drop fields from the context browser into almost any wizard field, which allows you to create dynamic content in wizards.
- The “Workflow Context” tab exposes values from the current instance of the workflow.
- The Inline Functions tab expose methods that you can use to perform operations like inserting the current date, performing calculations or manipulating text values.
- Item References are essentially “primary SmartObjects" that are associated with the workflow (such as the Leave Request for the Leave Request workflow).

K2 Application with the K2 Designer: Basics (Mastery Checkpoint)

Video

This is a checkpoint for the information covered in Part 1 of this module: building basic applications with K2 Designer. If you are attending a training session led by a K2 instructor, this is your opportunity to ask questions on any topics you might not yet understand.

These are the main concepts you should understand:

- Data
  - How to build a SmartObject that uses SmartBox as its data store
- Forms
  - Building very basic Views and using those Views on Forms
  - Basic Form Rules
- Workflow
  - How to build a very simple workflow in K2 designer
  - The difference between system tasks and user tasks
- How to use a K2 application
  - Starting a workflow
  - Opening and completing tasks

Knowledge-check questions

Q: Did your application work? If not, could you determine why not?
A: (discussion question)

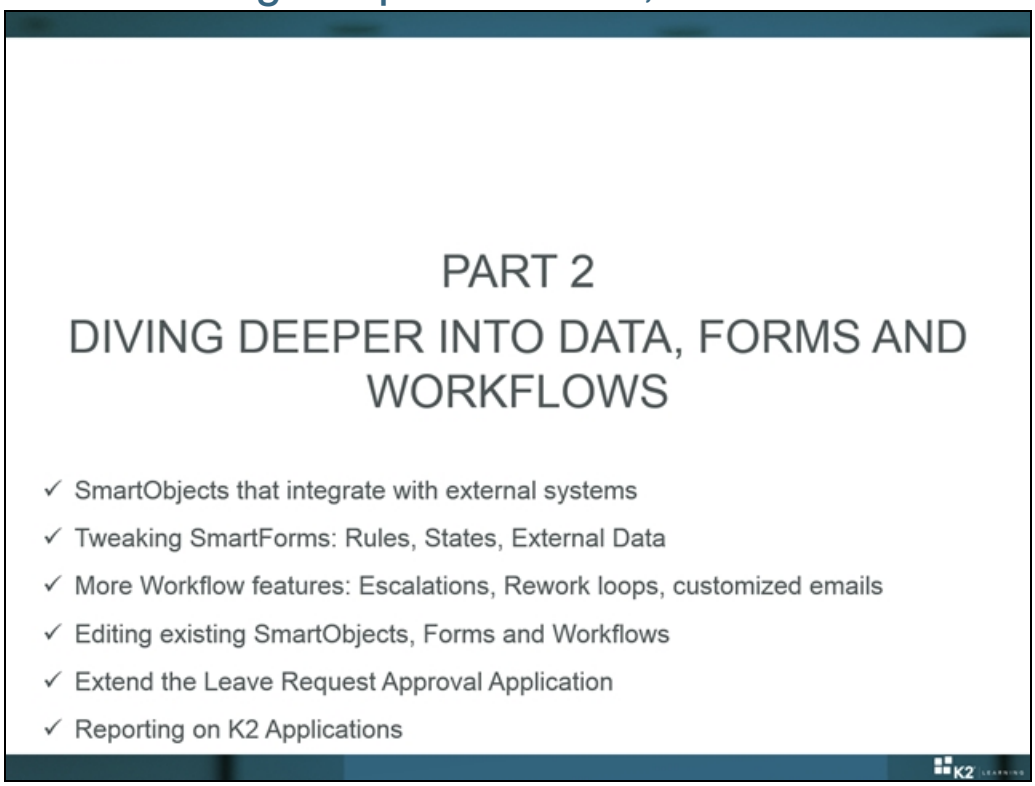
Q: Did you struggle to understand the differences between Data, Forms and Workflow?
A: (discussion question)

Q: Do you understand that K2 stores the Leave Request data internally in the SmartBox table created on the SQL database?
A: (discussion question)

Q: Do you understand the difference between “Views” and “Forms”?
Reveal answer**A:** Forms contain one or more Views. You can re-use the same View on different Forms. Users usually interact with Forms.

Q: Do you understand the concept of “Server Tasks” and “User Tasks” in a workflow?
Reveal answer**A:** Server tasks are tasks for K2. The workflow continues after the server task is completed. User tasks are tasks for humans. The workflow waits until the users have completed their tasks.

Part 2: Diving Deeper into Data, Forms and Workflows



Video

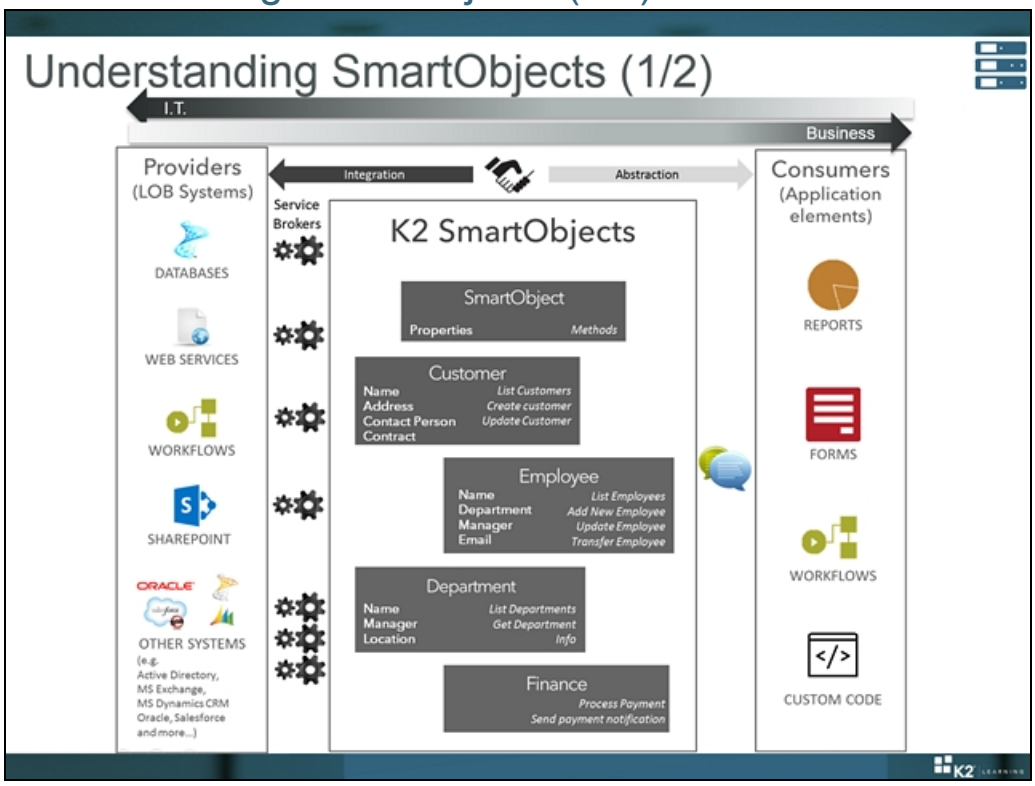
In Part 2 of this module we will dive deeper into the Data, Forms and Workflow components of K2 applications. We separate Part 2 into Data (with a exercise), then Forms (with an exercise) and finally Workflows (with an exercise). In the exercises for this module, you will expand on the basic version for the Leave Request Approval application that you created in Part 1, and add more functionality and features to the application.

You will learn:

- How to use SmartObjects to integrate with an external system
- A little more detail on working with SmartForms
- More detail on workflow components like:
  - Escalations
  - Rework loops
  - Customized emails and task notification messages
  - More advanced outcomes like multiple approvals
- How to report on K2 Applications

Note
Note: this module is not intended to cover K2 smartforms in-depth, so we will keep the SmartForms aspect light. If you will be using SmartForms, there is a separate training course that covers SmartForms in much more detail. We are using SmartForms since it is one of the easiest ways to build user interfaces without any third-party tools like SharePoint or Visual Studio.

Understanding SmartObjects (1/2)



Video

This topic is a refresher of the concept of SmartObjects, but notice that we have added an extra component to the diagram: the concept of Service Brokers.

You may recall that **SmartObjects** are the middle layer between **Data Providers** and **Data Consumers**. Consider the diagram in the slide for this topic. On the left, we have some Providers of Data (Data Providers could be a SQL Server table, Active Directory, a SharePoint List, web services or a number of other technologies). On the right, we have some Consumers of that Data (Data Consumers could be Forms, Workflows or Reports). SmartObjects are essentially an abstraction layer that makes it possible to integrate with many different systems as if they were logical business objects.

K2 uses something called "Service Brokers" which are technology-specific "connectors" that understand how to interact with particular systems. However, because of the way in which SmartObjects abstract these connections, the consumers only need to understand how to interact with SmartObjects and K2 takes care of the rest.

When you want to expose a certain Line-Of-Business system as SmartObjects (for example, exposing the entities in your organization's HR SQL database), you first have to configure an Instance of the Service Broker so that the broker can connect to the target system you want to expose. In this case, you may configure an instance of the SQL Server Service Broker and tell it where the SQL database is that you want to connect to.

Once the objects in the target system are discovered, you can create SmartObjects for those system objects and finally use those SmartObjects in Reports, Workflows and Forms.

In practice, the IT department (Admins and developers) are usually more involved with the left-hand side of things: registering and configuring service instances and potentially creating or generating SmartObjects, especially more complex SmartObjects. Business users are more involved in consuming the SmartObjects in their applications, such as using the logical “Employee” object on a Form.

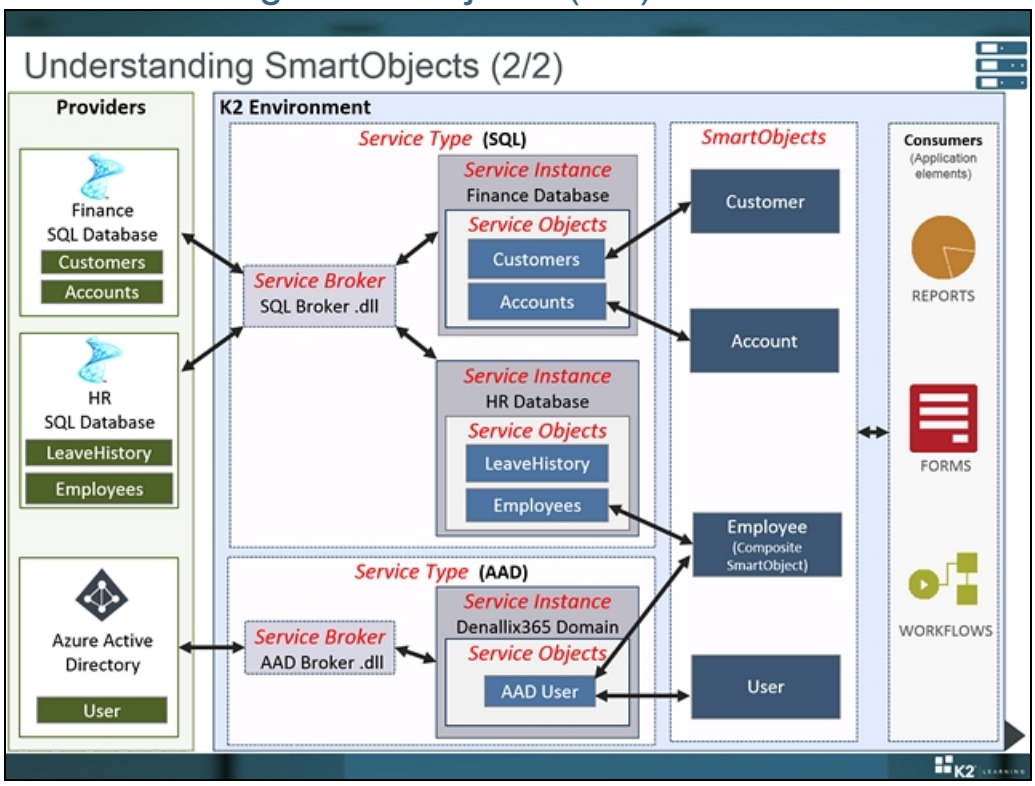
In the next topic, we will break down the SmartObject architecture a little further.

Summary

- SmartObjects are a "middle-layer" between Providers of Data (such as Databases and web services) and Consumers of Data (such as Forms and workflows)
- K2 uses something called “Service Brokers” to integrate with different systems

- The Service Brokers are usually specific to the technology being exposed. (i.e. there is a broker that knows how to talk to SQL databases, there is a different Service Broker that knows how to talk to SharePoint and so on)
- To interact with a particular provider, you would configure an Instance of the Broker with the necessary information to connect to the data source

Understanding SmartObjects (2/2)



Note
This topic contains fairly technical content. It is not critical to understand everything in the slide, as long as you are familiar with the basic architecture of SmartObjects.
The terms are important to understand if you want to know how to integrate specific back-end systems with K2 SmartObjects and will be creating SmartObjects to interact with other systems.

Service Objects, Service Types and SmartObjects are all components of K2 SmartObjects that allow K2 to connect to and interact with Data Providers and Data Consumers. Service Types are specific Data Providers such as Active Directory, Oracle Databases or Microsoft SQL Databases.

Sometimes you may need to configure an instance of a Service Type (this is called a Service Instance) because you have to provide a connection string so that K2 knows how to connect to the provider you want to interact with.

Service Objects are just an internal representation of the items that K2 discovered in the Data Provider. Service Objects are the Properties of the SmartObject.

Consider the diagram in the slide above. Here we have a SQL database for Finance. We have an instance of the SQL service that is configured to point to the Finance SQL database. Registering the service instance will discover the available tables, views and stored procedures in the target database and generate Service Objects for those artifacts. Then, we can either auto-generate SmartObjects for the Service Objects or manually create SmartObjects using K2 design tools.

Next, we add another Service Instance of the SQL service type to expose an HR SQL database as SmartObjects. The same discovery procedure follows the entities in the target database and are exposed as Service Objects. We could now generate or manually create SmartObjects for those Service Objects if we wanted to, but we don't have to.

Finally, we use an instance of the Azure Active Directory (AAD) service type to expose user data that lives in our Azure Active Directory domain for Denallix. Again, this creates Service Objects for the entities in the AAD system.

We can combine service objects to build up Composite SmartObjects: in this case, we are combining data from the Azure AD store and the Employee database to create a logical "Employee" SmartObject that combines data from both systems and represents it as a logical business object to the consuming Forms, Workflows and Reports. (If you are

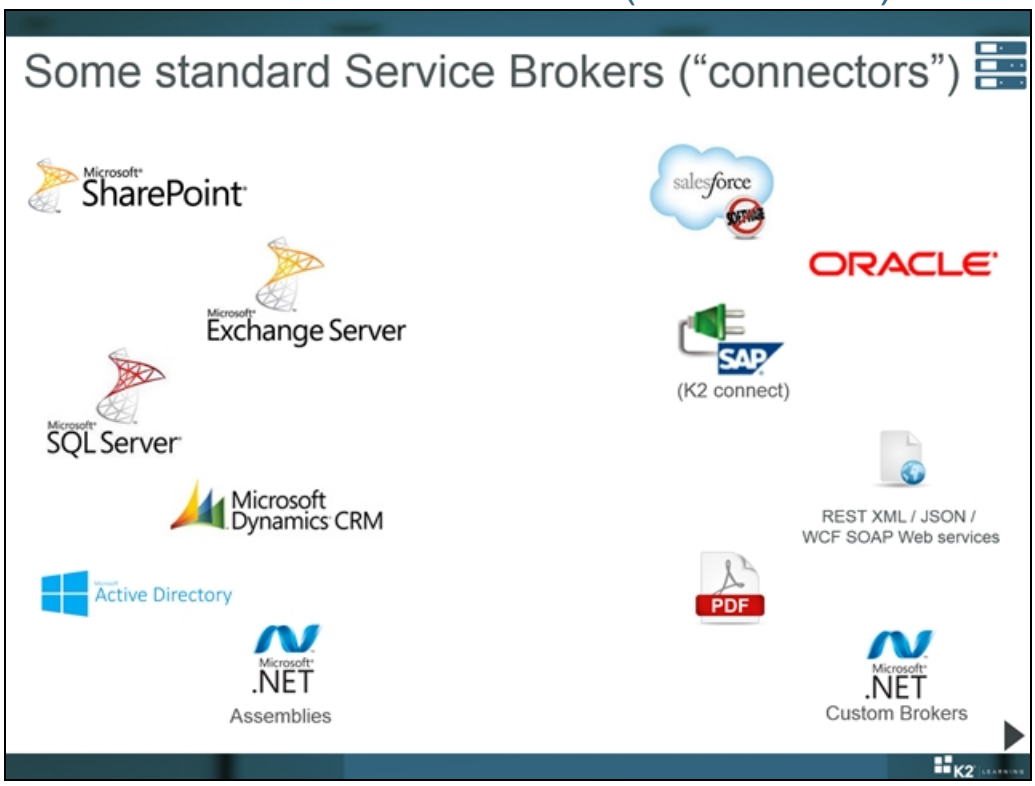
familiar with SQL, composite SmartObjects are similar to JOIN statements for SmartObjects except that the data could come from different systems).

What is important is that Data Consumers don't know anything about the underlying systems. As far as they are concerned, they "talk to" an abstracted yet consistent set of business entities and K2 takes care of the integration behind the scenes.

Summary

- Service Instances are used to point a Service Type to a specific data source, for example registering an instance of the SQL Server Service Type to point it to the Finance Database.
- Registering a Service Instance will discover the available artifacts in the target system, such as SQL database Tables and Stored Procedures.
- You can then manually create or generate SmartObjects for the artifacts in the target system.
- You can have multiple instances of the same Service Type that each point to a different target system.
- You can combine data from multiple Service Objects to create Composite SmartObjects that combine data from different systems.

Some Standard Service Brokers ("connectors")



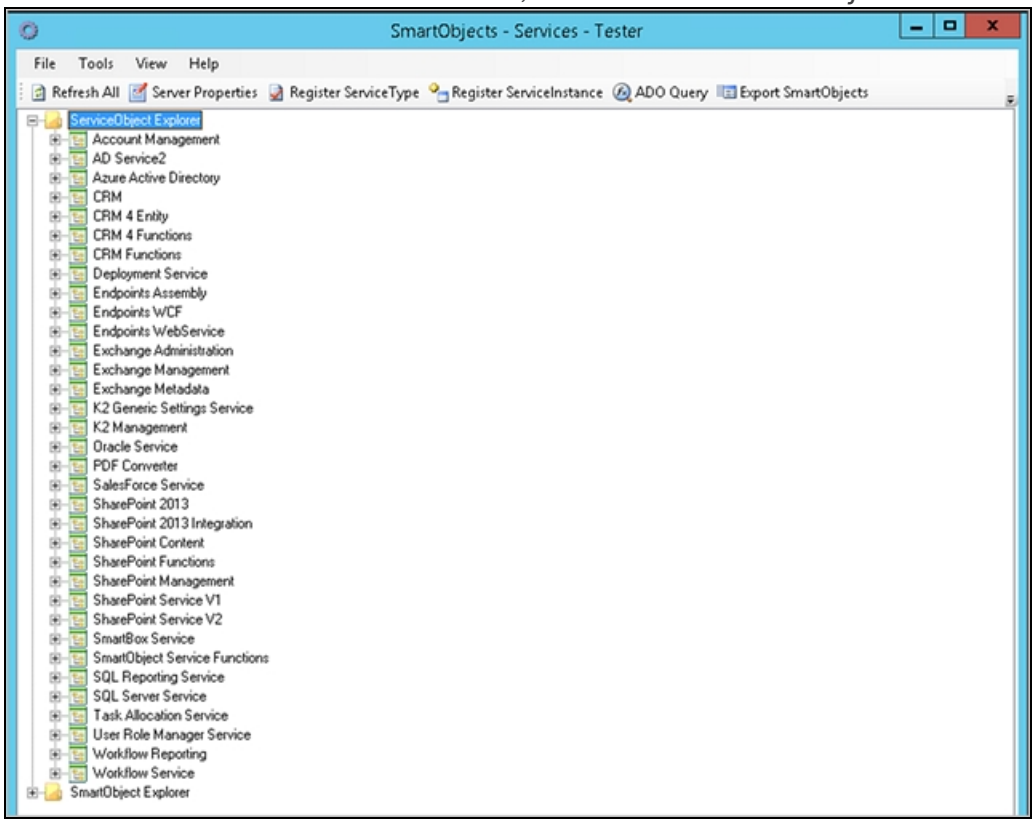
Video

K2 comes provided with several standard Service Brokers to provide integration with third-party systems as well as K2 itself. There are several brokers specific to Microsoft technologies (such as Microsoft Exchange, Microsoft SQL Server, Microsoft SharePoint, Microsoft Active Directory, Microsoft Dynamics CRM and others) and several for other third-party technologies (such as Oracle databases, Salesforce, SAP, Web Services, PDF output and more). Some of these brokers are not bound to a particular vendor, such as the Services Endpoints brokers which can be used to interact with web services, regardless of the vendor that provides those services.

It is possible to create custom brokers to expose other systems to K2, but writing these brokers is a developer task. Once the custom Service Broker is created and registered with a K2 environment, users can start to use that Broker to register Service Instances and create SmartObjects.

The screenshot below shows a selection of the available brokers as shown when using the K2 Service Tester utility. You will learn how to use this utility in a later exercise, for now just see how you can determine what brokers are enabled in your environment. Note also that different versions of K2 might have different brokers provided as standard.

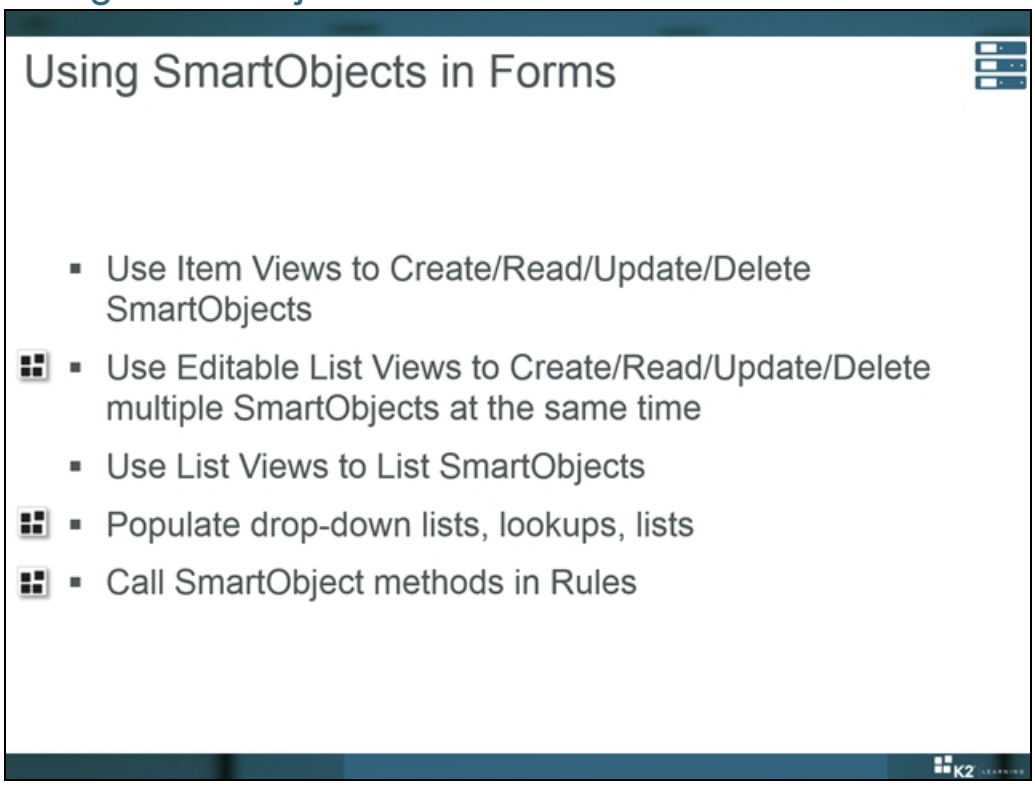
Some standard services included with K2, as shown in the SmartObject Service Tester utility



Summary

- K2 comes provided with several standard service brokers to provide integration with third-party systems as well as K2 itself.
- There are several brokers specific to Microsoft technologies and several for other third-party technologies.
- Some brokers like the Endpoints Services and WCF brokers allow you to interact with web services.
- It is possible to create custom brokers to expose other systems to K2 as well.

Using SmartObjects in Forms

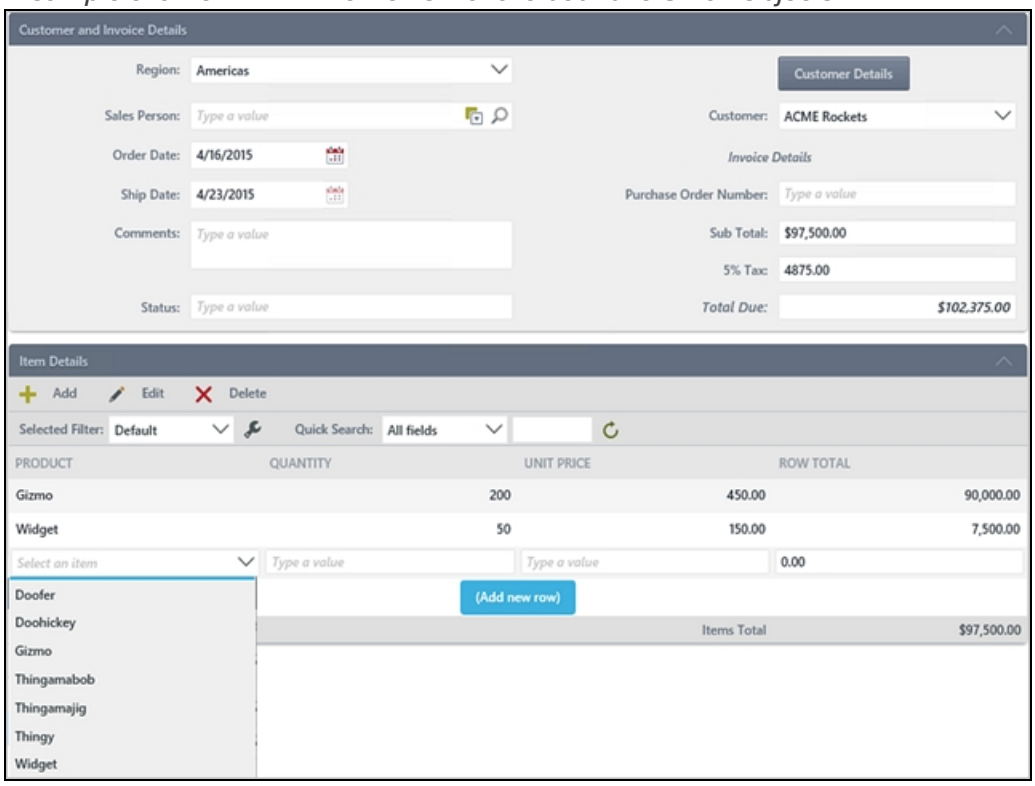


Video

SmartObjects are the main mechanism in which SmartForms integrate with other systems. Normally, a View would be bound to a specific SmartObject, and then you can add Fields and Controls onto the View to work with that SmartObject.

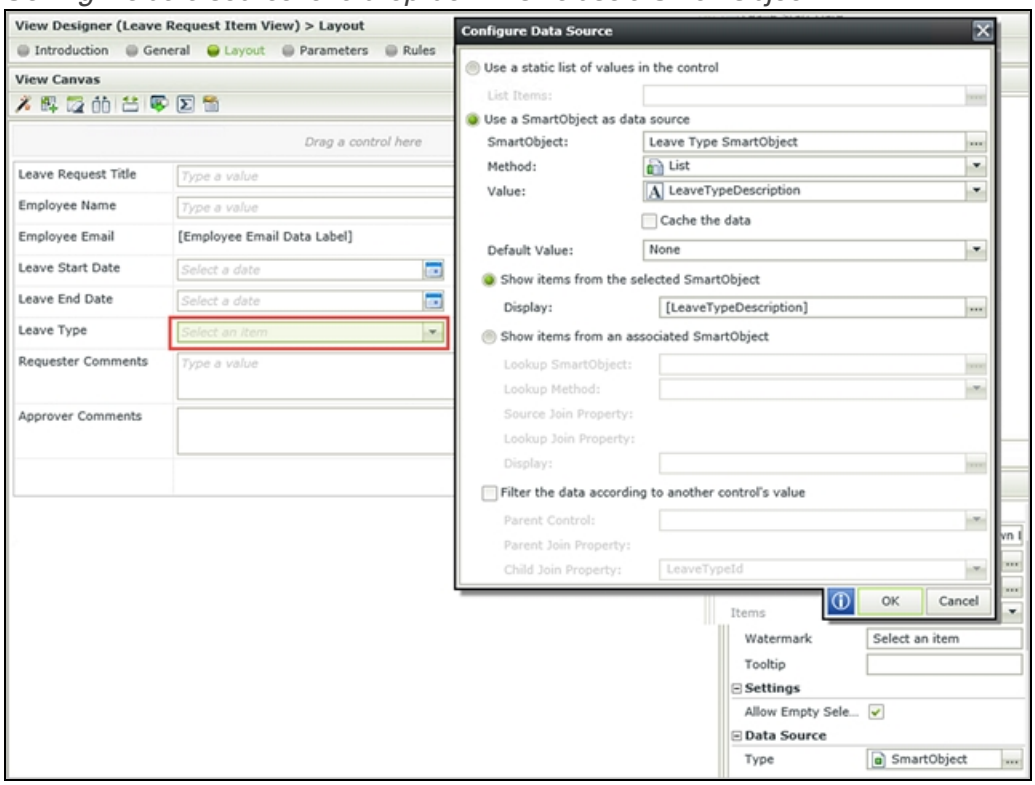
You can use a SmartObject as the data source for an Item View (very common in a scenario where users capture data for a single record), the data source for a List View (common in scenarios where you want to show a read-only list of multiple records), or the data source for an editable list (where users can capture data for multiple records at the same time, common in a data-entry type scenario).

A sample of a Form with two views that are bound to SmartObjects



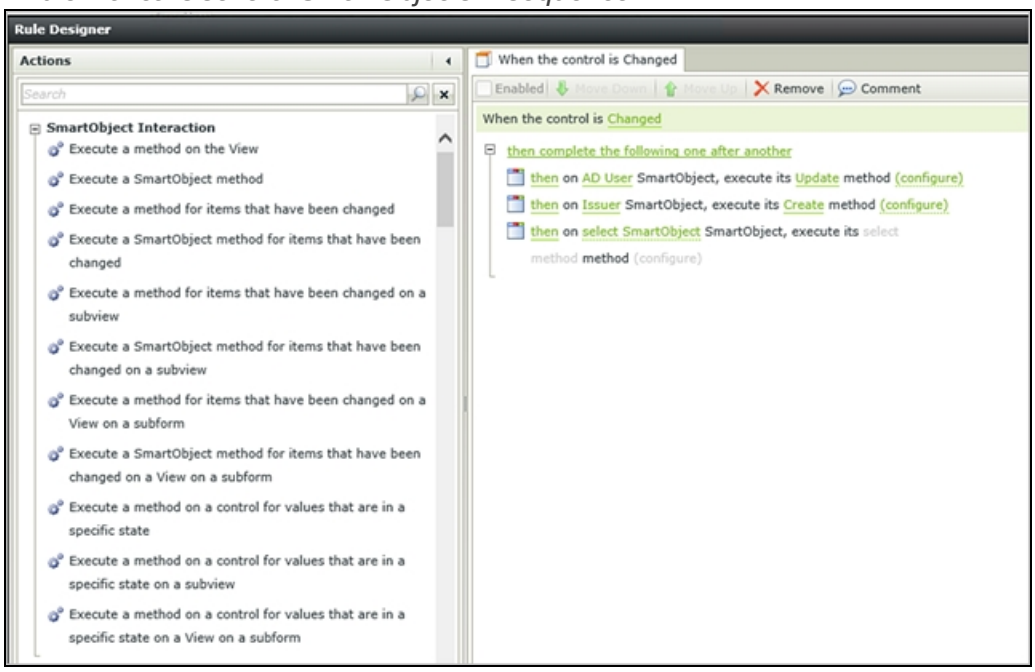
You can also use SmartObjects to populate things like drop-down lists, search boxes, list controls and so on, for example, giving users a choice from a drop-down list where the available values in the drop-down list come from a SQL database.

Setting the data source for a drop-down list to use a SmartObject



You can also define Rules that call SmartObject methods. The rules are not restricted to only the SmartObject used in that View, you can call any available SmartObject method in a Rule.

A rule that calls several SmartObjects in sequence



Summary

- SmartObjects are the data access mechanism for SmartForms
- You can bind SmartObjects to Item Views and List Views
- You can use SmartObjects to populate things like drop-down lists, search boxes, list controls, etc.
- You can call SmartObject methods in SmartForms rules

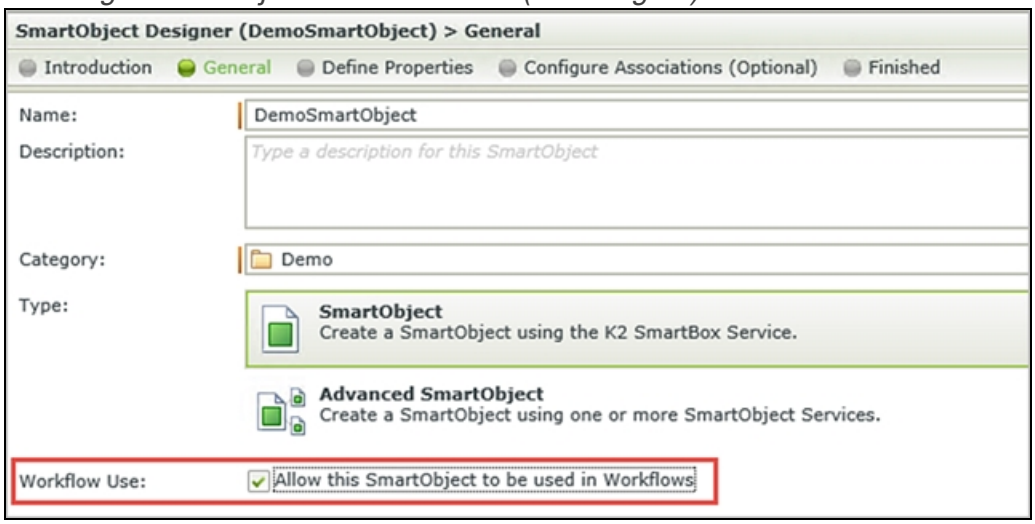
Using SmartObjects in Workflows

Video

You can also use SmartObjects in workflows to interact with other systems. When using the web-based workflow designer, note that SmartObjects will not be available in workflows unless the “use in workflow” option was selected when creating the SmartObject in K2 Designer. (The “Use in workflow” option ONLY applies to workflows designed with the web-based workflow designer. It does not apply to workflows designed in K2 Studio or K2 for Visual Studio. Those designers can call any SmartObject.)

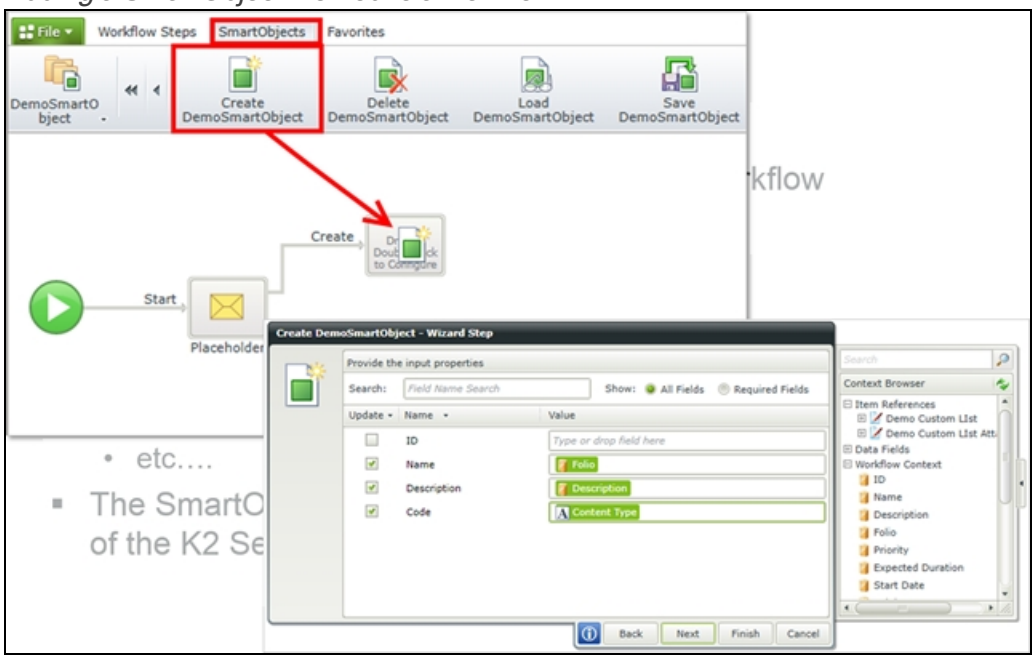
Note
When a workflow calls a SmartObject, note that the method will be called in the security context of the K2 Service Account, so if you are updating a database with auditing enabled, for example, the value would appear to be updated by the K2 service account user, since this is the security context in which the SmartObject was called.

Enabling a SmartObject for workflow use (K2 Designer)



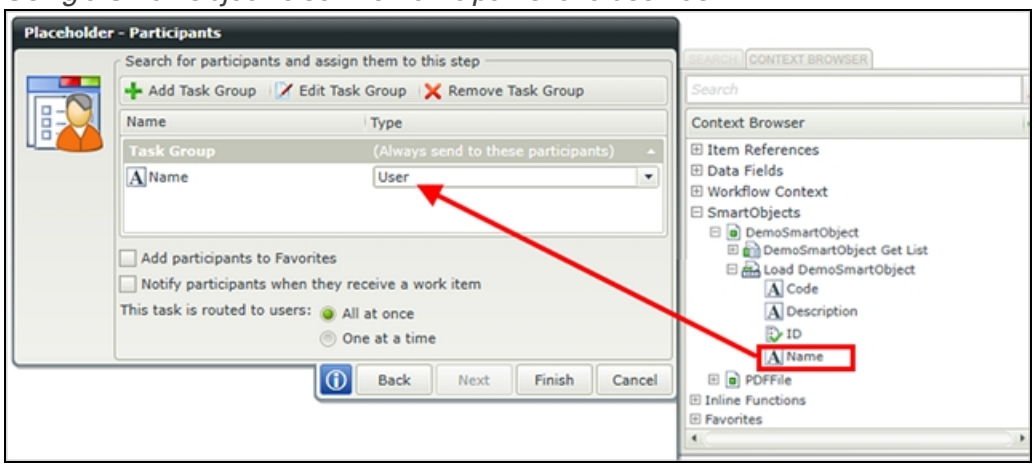
Once the SmartObject is made available to workflows, you will see the available SmartObject methods in the workflow designer ribbon bar, and you can drag, drop and configure the methods just like any other workflow wizard.

Adding a SmartObject method to a workflow

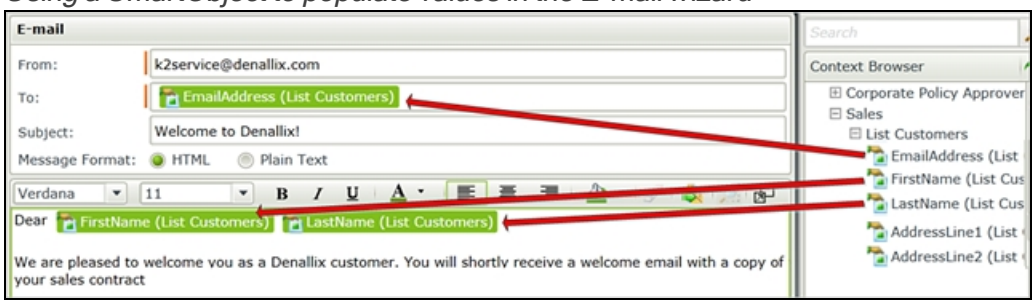


You can also use the SmartObjects from the Context Browser, and use the result of a SmartObject in something like the participants for a User Task or in a wizard.

Using a SmartObject to set the Participants for a user task



Using a SmartObject to populate values in the E-mail wizard



Summary

- Enable the “use in workflow” option to expose SmartObjects to workflows.
- You can also use return properties from a SmartObject in things like participant rules, wizards, inline functions, etc.
- When you call a SmartObject in a workflow, the SmartObject is executed in the security context of the K2 Service Account.

SmartObjects (Mastery Checkpoint)

Video

This is a checkpoint for the information covered in Part 2 of this module: SmartObjects. If you are attending a training session led by a K2 instructor, this is your opportunity to ask questions on any topics you might not yet understand.

These are the main concepts you should understand:

- How SmartObjects connect to other systems. (If more technical, you should at least know what “Brokers” and “Service Instances” are.)
- What SmartObject Service Brokers are and that brokers are technology-specific.
- How to use SmartObjects in SmartForms and in Workflows.

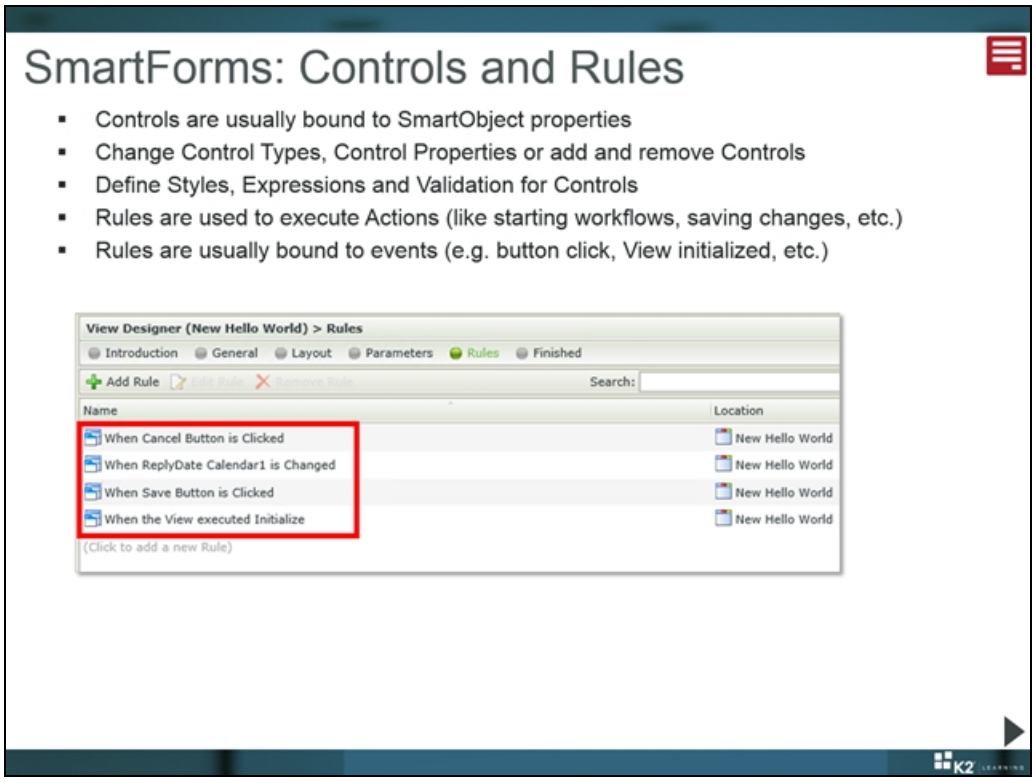
Knowledge-check questions

Q: True or False: A SmartObject can only be used in workflows.
Reveal answer**A:** False, they can be used in Forms, Workflows, Reports and custom code as well.

Q: What is the difference between a SmartObject and a Service Broker?
Reveal answer**A:** A Service Broker is a technology-specific “adapter” that is used by SmartObjects to interact with a particular system. A SmartObject is the logical business object that exposes one or more providers (of data) to consuming applications like Forms and Workflows.

Q: True or False: You can use the Oracle Service Broker to connect to a Microsoft SQL Database
Reveal answer**A:** False, because these brokers are technology-specific. You would use the SQL Server Service Broker to connect to a SQL database.

SmartForms: Controls and Rules

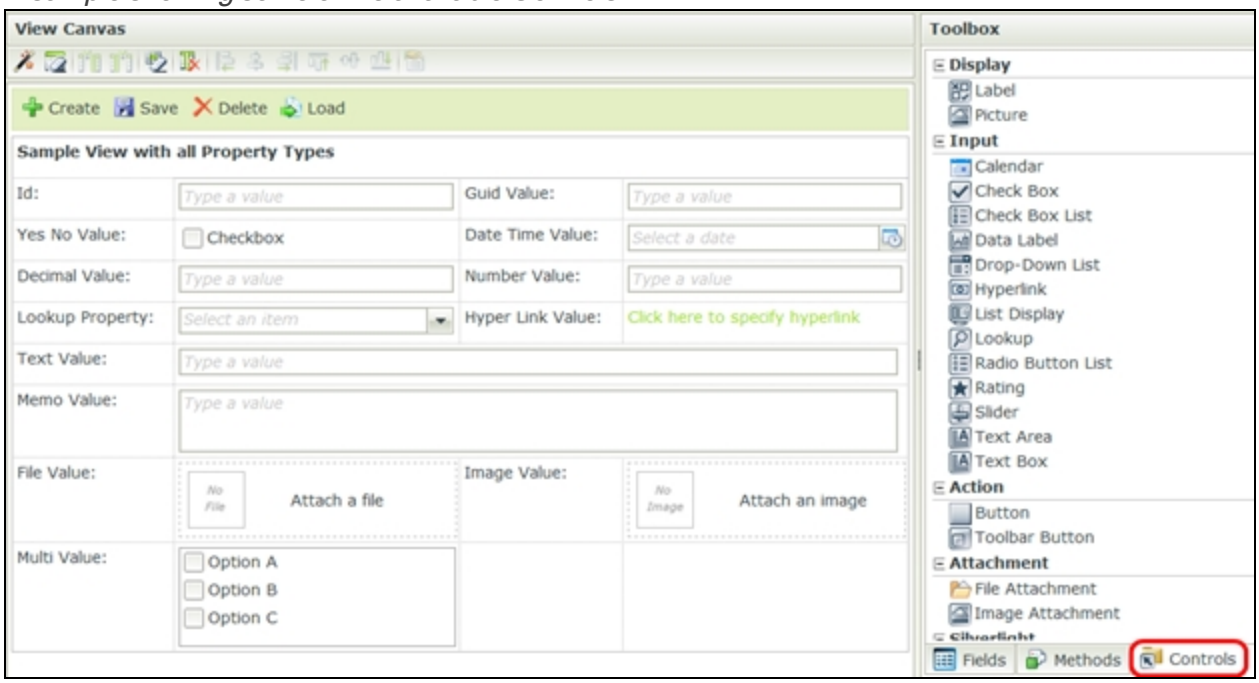


Video

Controls

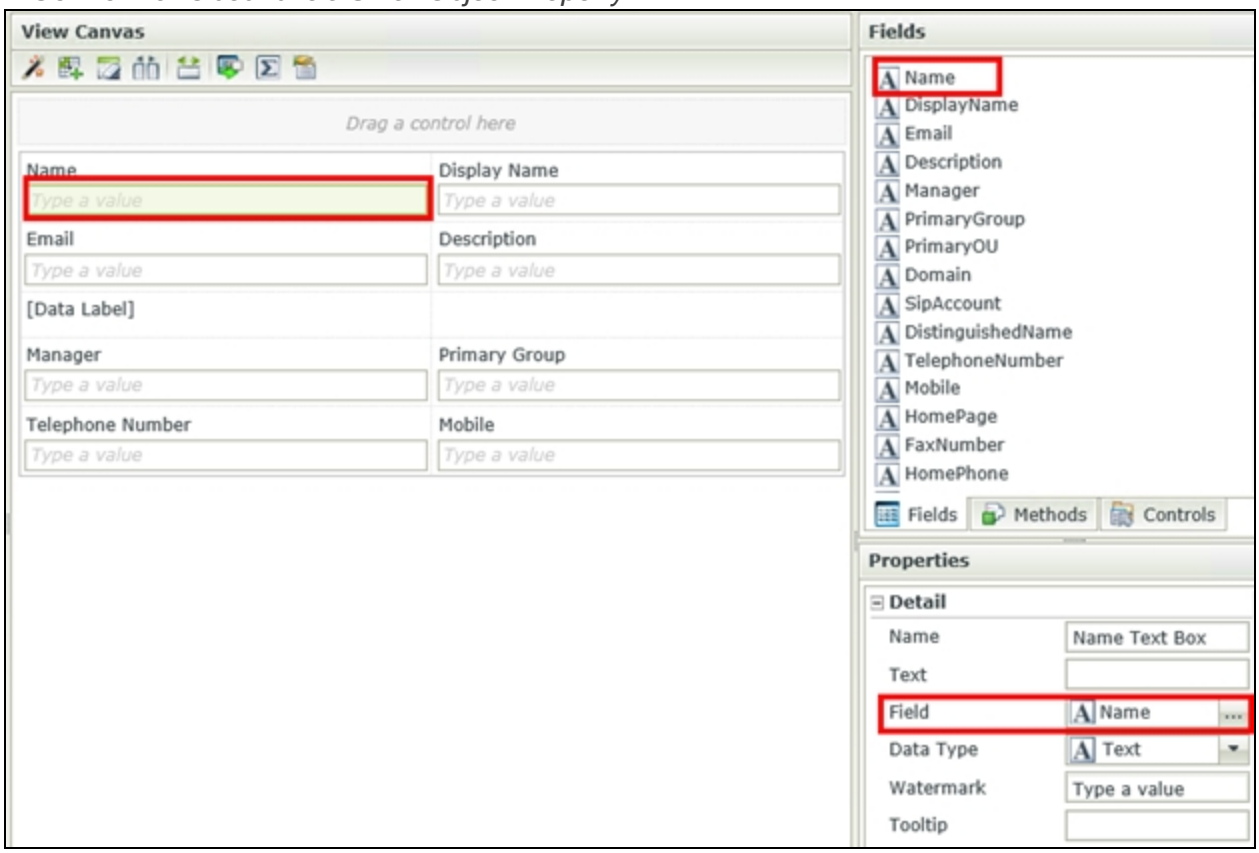
Controls are used on Views and Forms to display data or values and to capture user input. There are a range of controls available in the Toolbox on the right-hand side of the screen: the screenshot below illustrates a selection of the available Controls.

A sample showing some of the available Controls



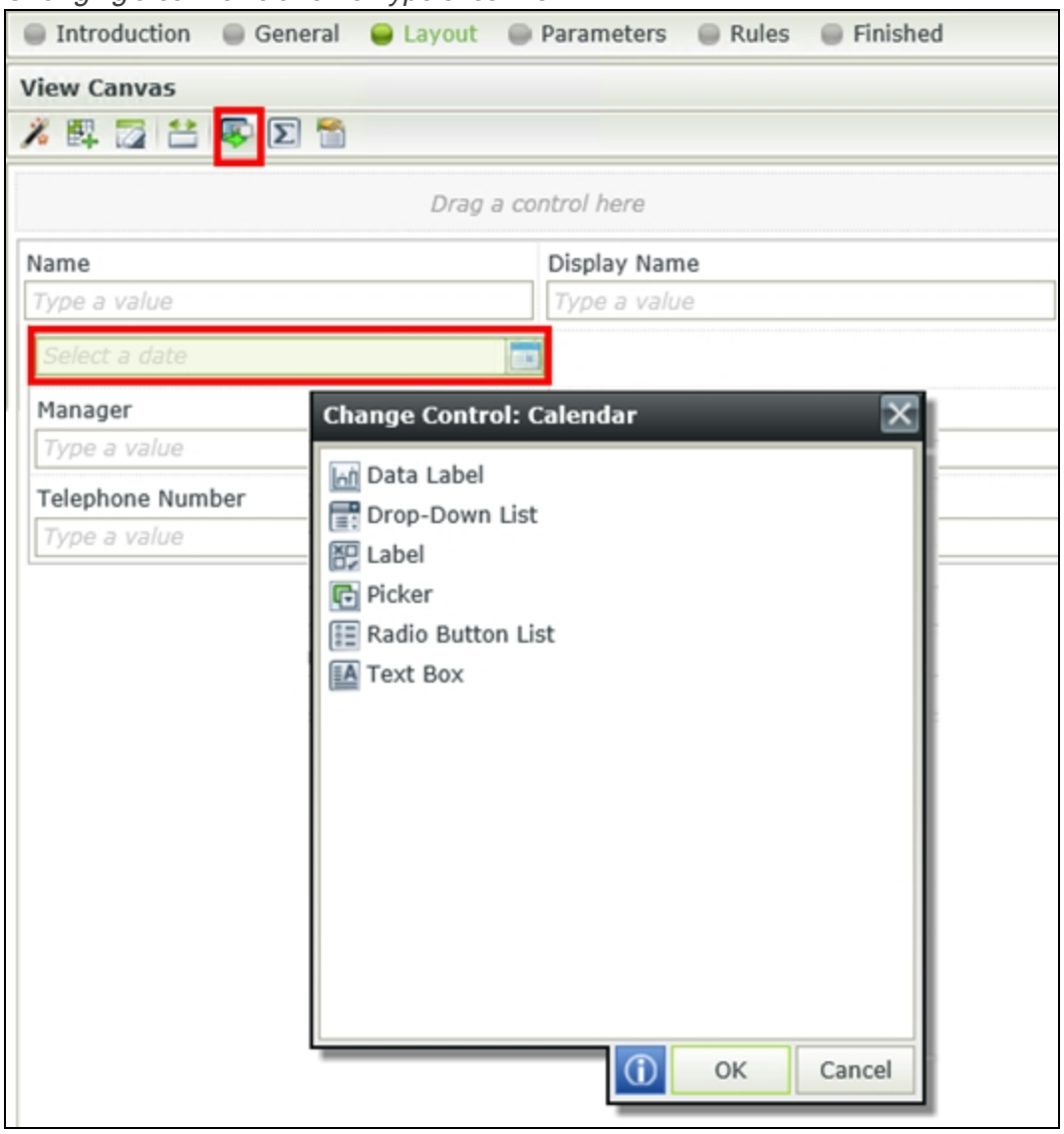
Controls are usually bound to a SmartObject property, and can also be used to display data from another SmartObject (for example using a drop-down list control to show a list of values returned by a SmartObject).

A Control that is bound to a SmartObject Property



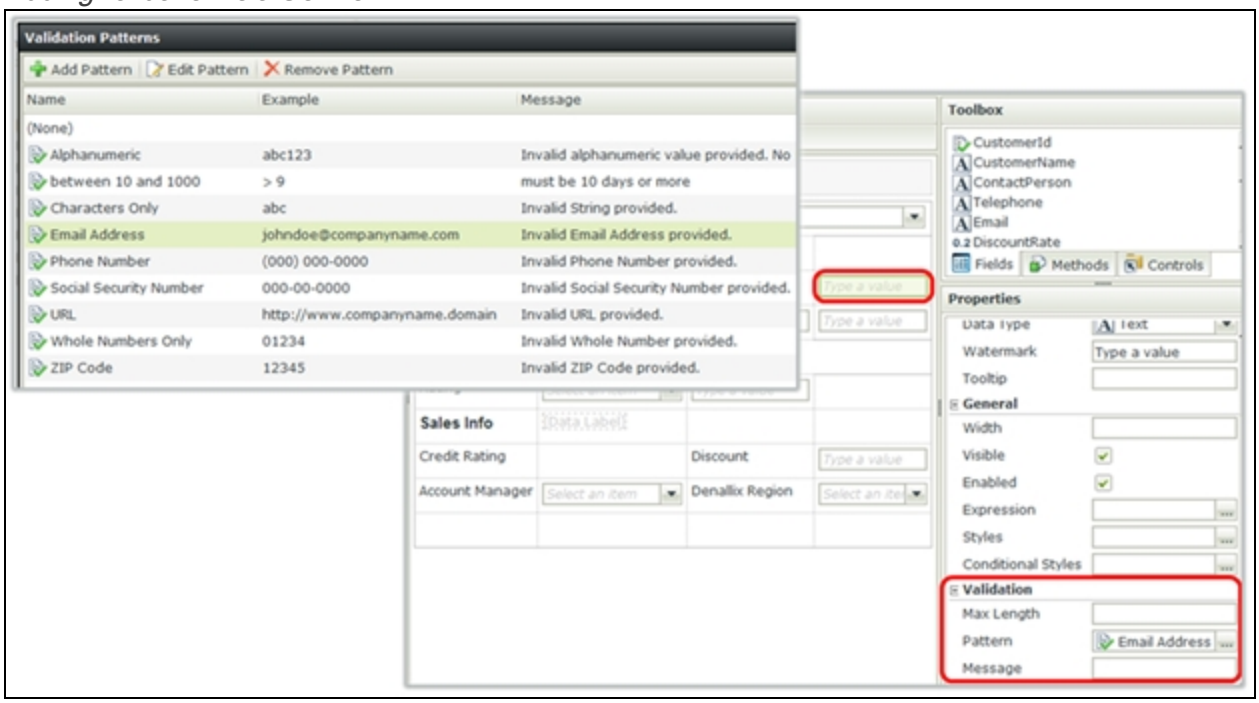
When you generate a View or drag and drop fields from a SmartObject onto a View or Form design canvas, K2 will automatically assign an appropriate type of control for the data type of the field. For example, if you drag a date/time field onto the design canvas, K2 will insert a calendar control onto the design canvas because this is the most common control used to input a date/time field. However, it is possible to change the type of a control at design time using the toolbar, as shown below.

Changing a control to another type of control



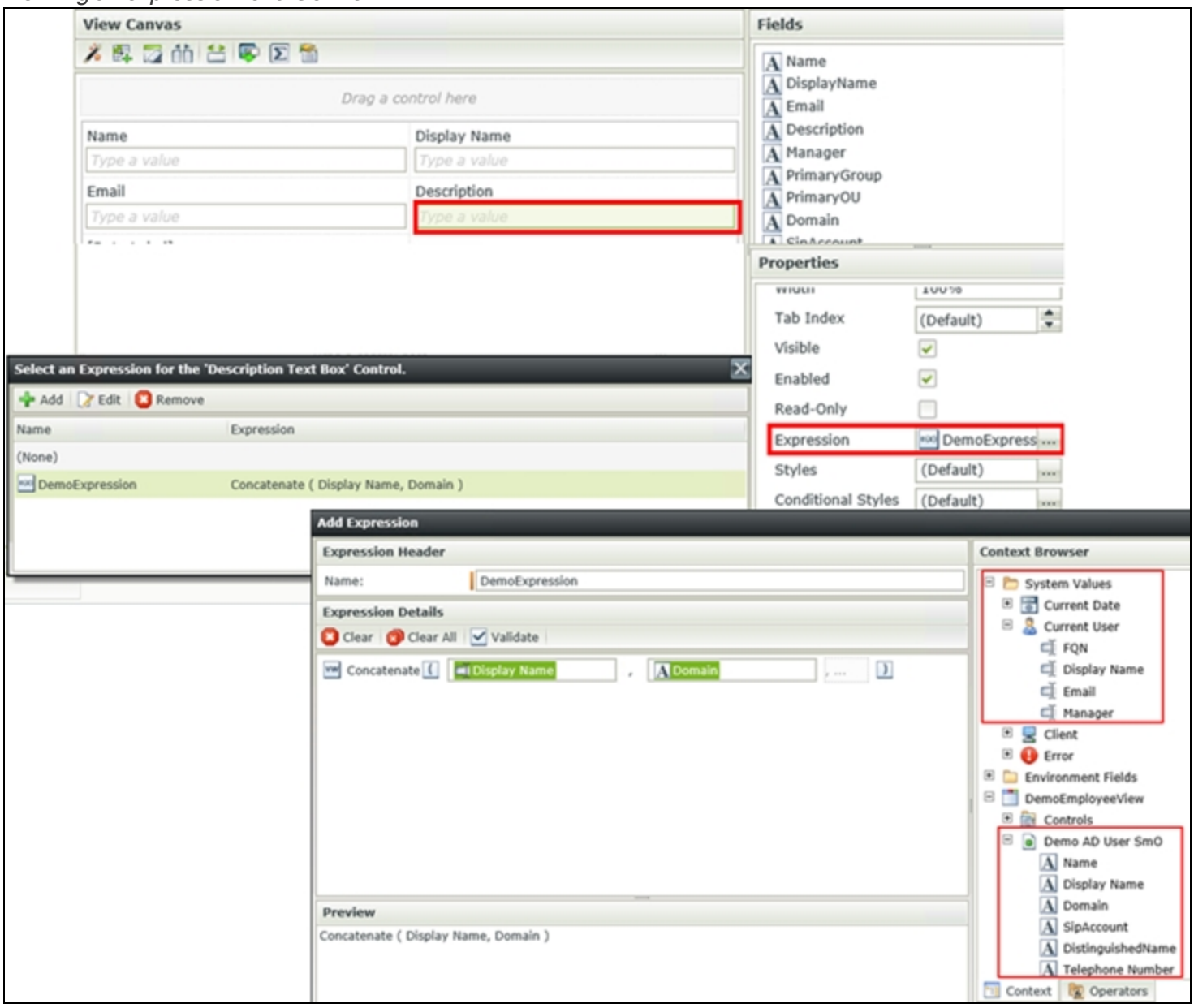
You can set Validation properties on Controls that can match the value entered in the control with a particular pattern or format. For example, you can select the **Email Address** pattern to verify that the value entered by the user is in the correct format for an email address. You can also define the maximum length allowed for input as well as a custom message if the validation is not successful, as well as defining custom patterns for other types of validation expressions.

Adding validation to a Control



Expressions can be used to calculate a value for a Control. For example, you may want to concatenate the values of two fields together and display them in one Control. To achieve this, use the **Expression** property and define an expression using the available **Operators** in the Context Browser. You can also include values from the current environment (such as the current username or current date) that are available in the System Values group in the Context Browser, as shown below.

Defining an expression for a Control



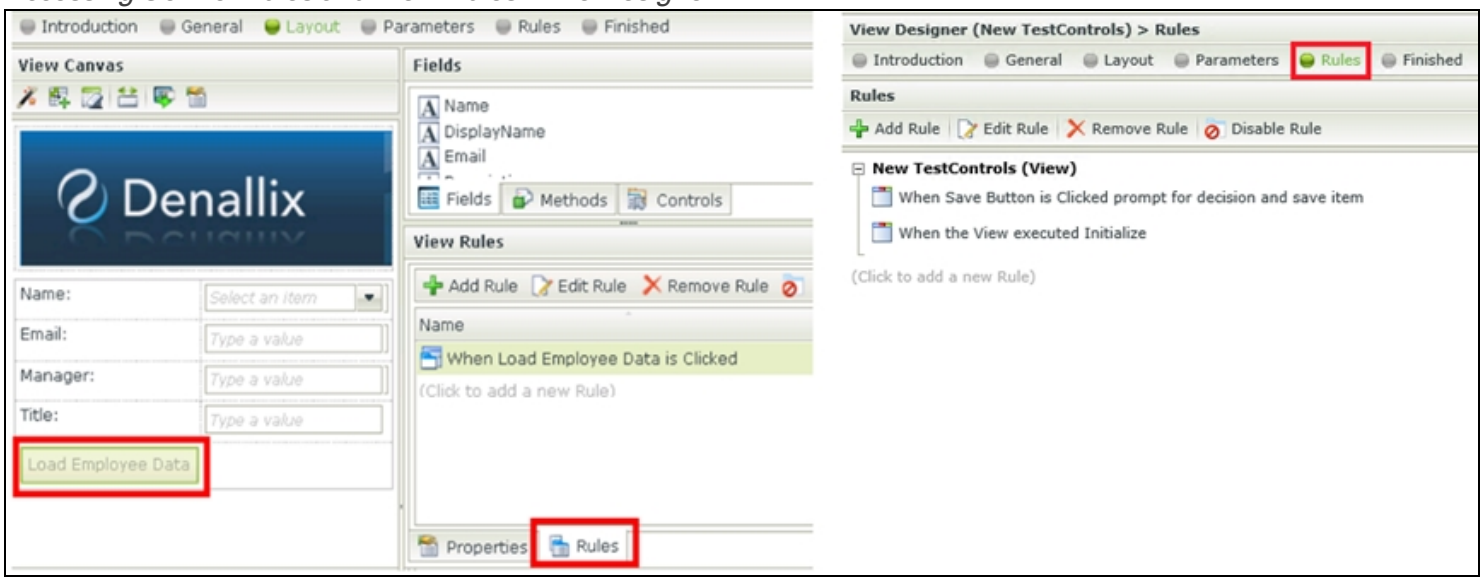
Controls can raise Events that are handled with Rules. This allows you to build some processing logic into a View when something happens on a control, for example refreshing the View content if a specific control is changed by the user. Different types of controls will have different events.

Rules

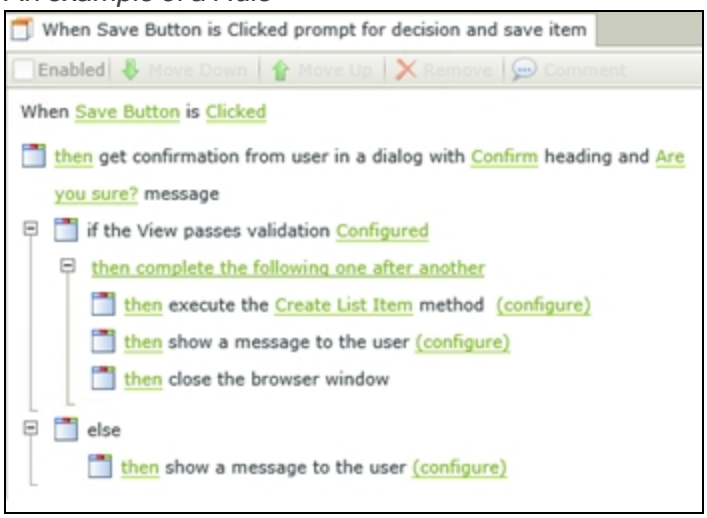
Most user interfaces require some kind of programming or processing support, for example saving updates to a record into a database when the user clicks the “Save” button, or starting some workflow when the user “Submits” a Form. SmartForms use Rules to perform these “programming” actions. (We use “programming” in quotes because there is actually no coding required when implementing the processing tasks. Instead, the form designer will use graphical, wizard-based configuration tools to assemble the programming logic.)

The screenshot below shows how to access the Rules property panel for a control and the Rules for a View or a Form. A control, View or Form may have multiple Rules, and most often each Rule is bound to a particular Event (but they don't have to be).

Accessing Control Rules and View Rules in the Designer

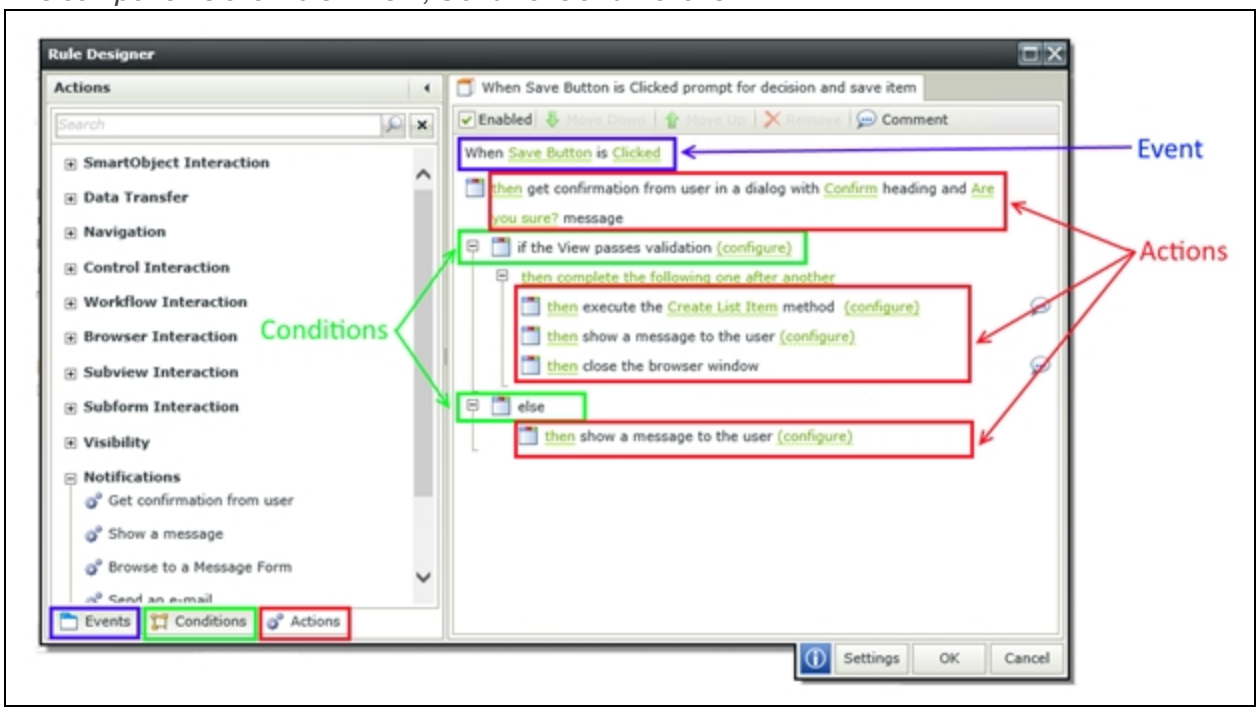


An example of a Rule



Rules are made up of three components: Events, Conditions and Actions.

The components of a Rule: Event, Conditions and Actions



Events

Events define WHEN the rule should “fire”, and are normally associated with an event on a Control, View or Page. Examples of events include ”when a button is clicked”, “when a control’s value changes” or “when the View is initialized”. In this example, we want the rule to fire when a control (a Button) on the View raises an event (Clicked). Note that Rules do not have to be bound to Events, and you can define "eventless" rules. Those rules are normally called by another rule instead of an Event. Depending on the Control you clicked or the type of View (List View or Item View) being designed, you may have different Events available in the Events list. To help speed up your development time, the most common Rules for the control you selected are also displayed under the Templates heading. These templates are just pre-defined rules that you can select from, or you can define rules from scratch using the available Events in the wizard.

Conditions

Conditions control IF a Rule is allowed to execute. These conditions are normally some kind of true/false result, and as with the Events designer, you will have a range of pre-defined conditions to choose from. You can also use the Advanced Condition configuration tool to define a more advanced condition for the Rule. By default, a Rule is set to execute “always” (in other words, if you do not define a condition, the rule will always continue to the Actions part).

In the example Rule above, we defined a condition so that the Actions would only execute if the Form passed validation.

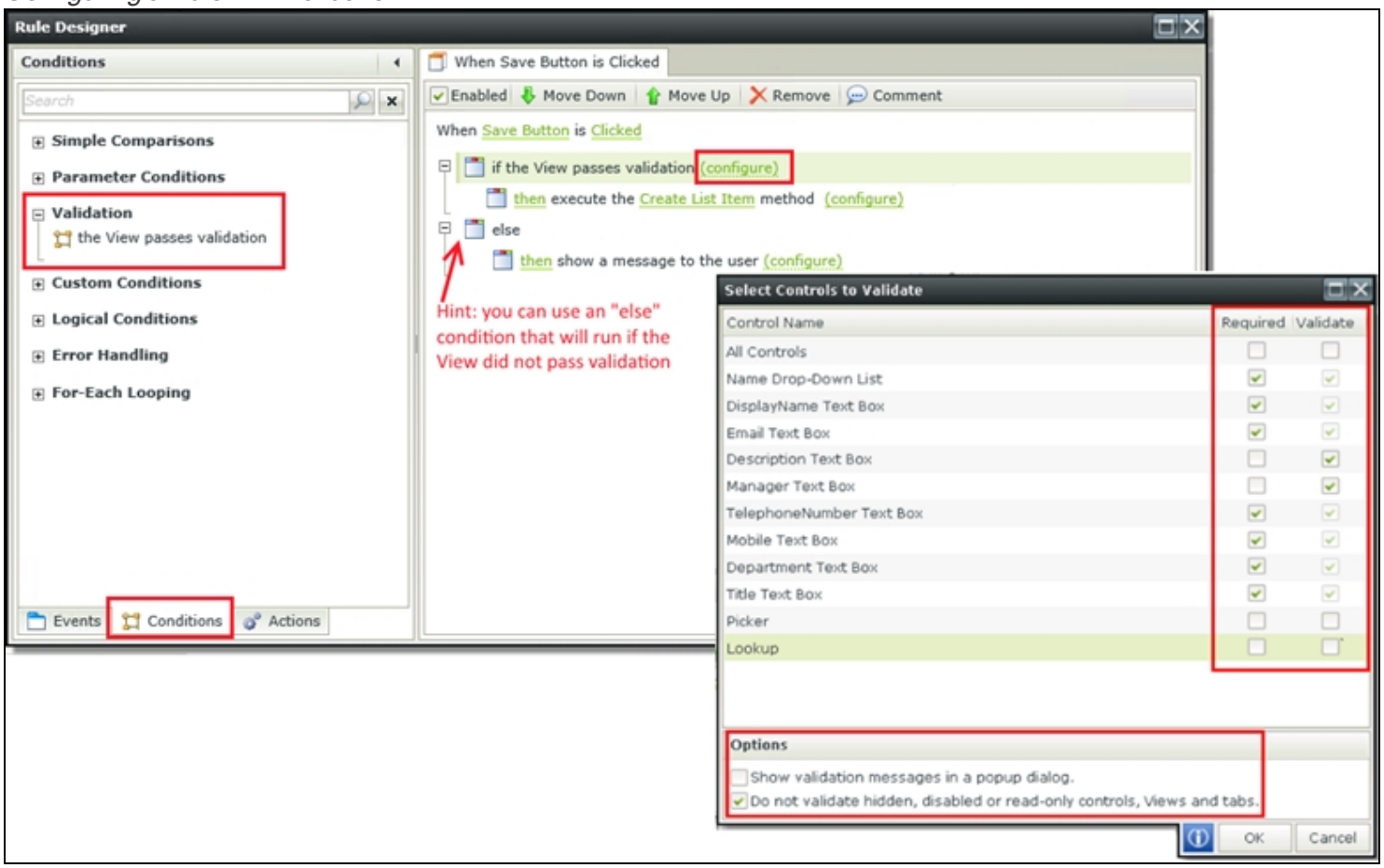
Actions

Actions define WHAT the rule does, in other words the work that is performed by the Rule. As with the other Rule components, when defining a Rule you will have a selection of operations to choose from and you can select multiple operations in the same action. Rules always have to have at least one Action, but could contain multiple Actions.

Configuring Validation

When configuring a "passes Validation" condition for a Rule, you can select which controls should be validated, where the validation error should be shown and whether or not to ignore hidden controls. Additionally, you can add an "else" condition which will execute if the Form or View did not pass validation. The screenshot below shows how this is implemented.

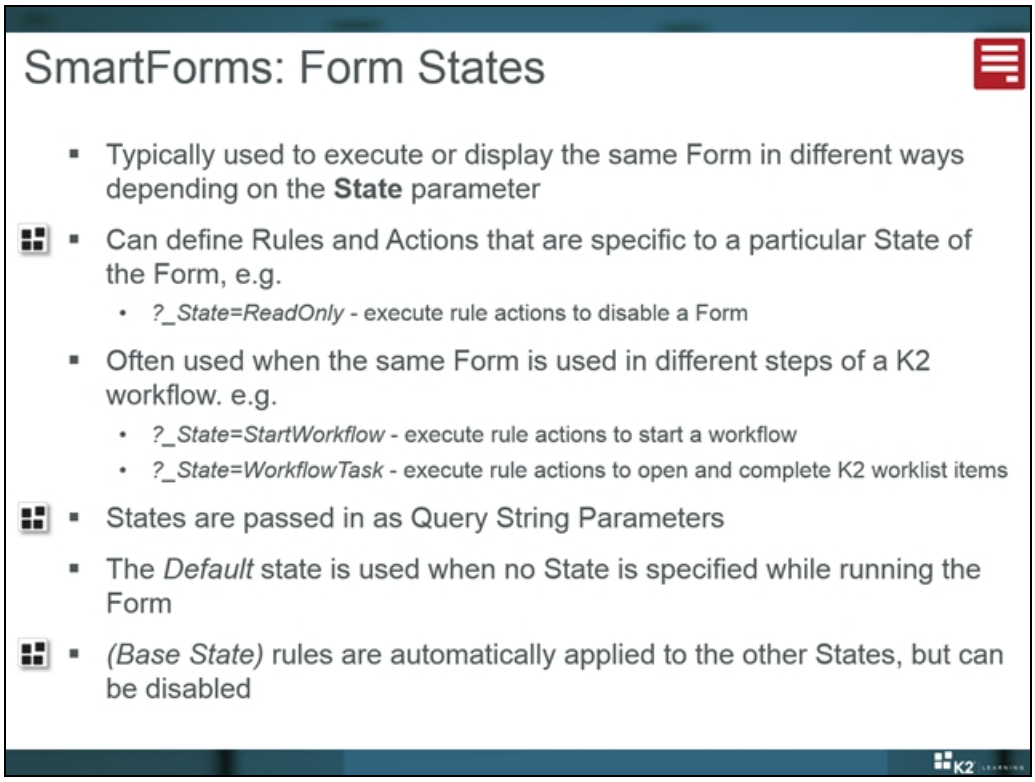
Configuring a Rule with Validation



Summary

- Controls
  - Can be changed to different control types
  - Are usually bound to SmartObject properties (but don't have to be)
- Rules
  - Are the "programming" logic in SmartForms
  - Can contain an Event ("when") a condition ("if") and one or more actions ("do this")

SmartForms: Form States



Video

We want to talk briefly about States in SmartForms because you will be using them in the next exercise. Although “States” are a SmartForms concept, the principle applies to other User Interface technologies as well, because a Form behaves differently when you want to start a workflow vs. when you are “in” a workflow and need to open and complete a task.

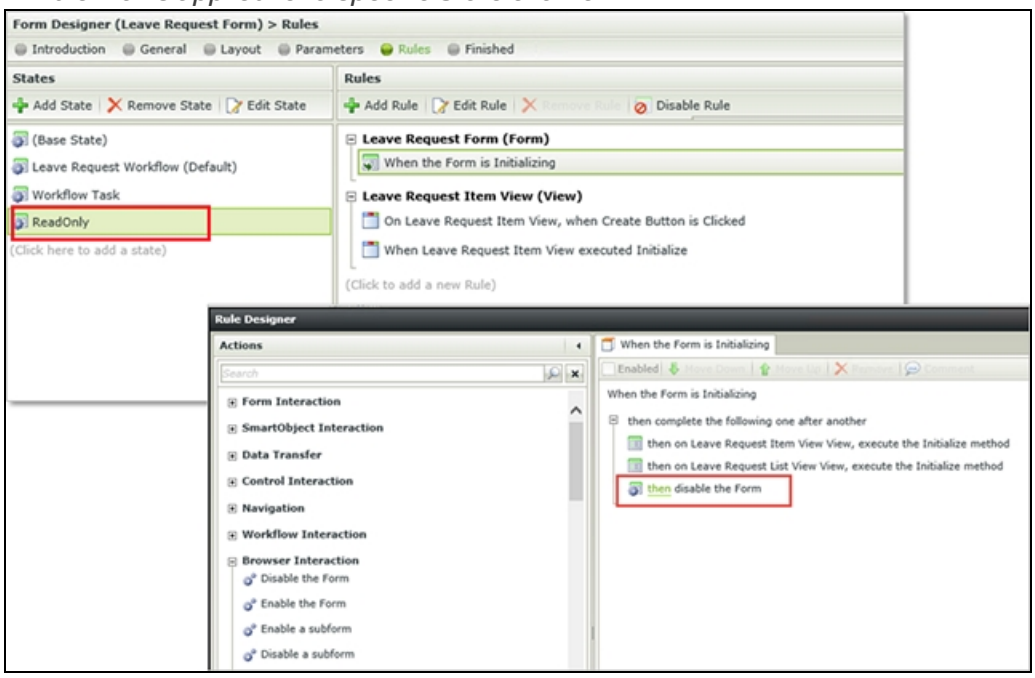
States are a way to re-use the same Form but apply different Rules depending on the State that the Form is currently in. Here is an example: suppose you have a complex Expense Claim approval form that is used in an Expense Claim Approval workflow. You would like to use the same Form to start the workflow and during the subsequent Approval tasks, but the Rules on the Form are slightly different depending on whether the Form is being used for Submission or for Approval.

This is where Form States are useful. You can define multiple Form states for the same form, and then configure additional rules that are specific to the current form state. For example, if the Form State is “submission” then enable all the input controls on the form and when the submit button is clicked, create the expense claim record and start the approval workflow. If the Form State is “Approval”, then open the worklist item with the Form Serial number, and disable all the input controls on the Form except for the manager’s approval decision drop-down menu.

Tip
The State of a Form is usually passed in as a Query String parameter _State=[StateName]

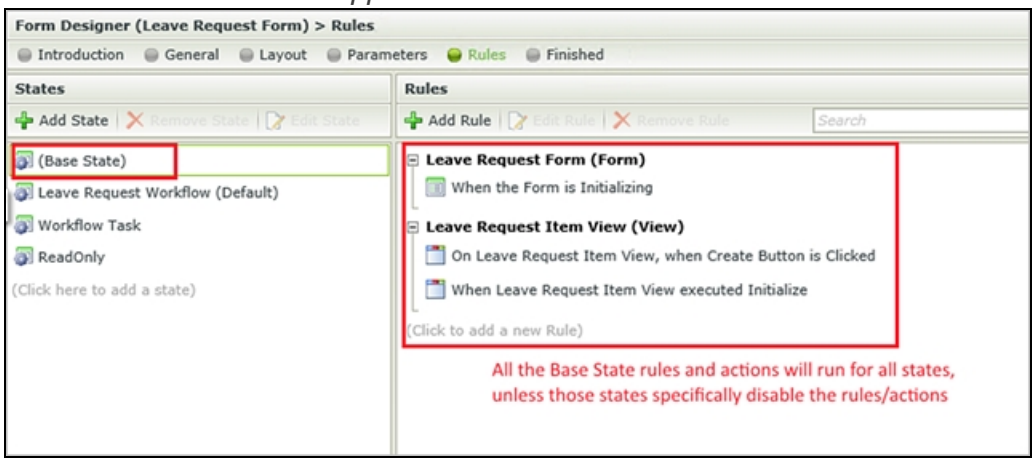
States are usually used in conjunction with Rules, so you would define additional States for a Form on the Rules wizard page, as shown below. For example if the state parameter is “ReadOnly”, then run a Rule Action to disable all the controls on a Form.

A Rule that is applied for a specific State of a Form



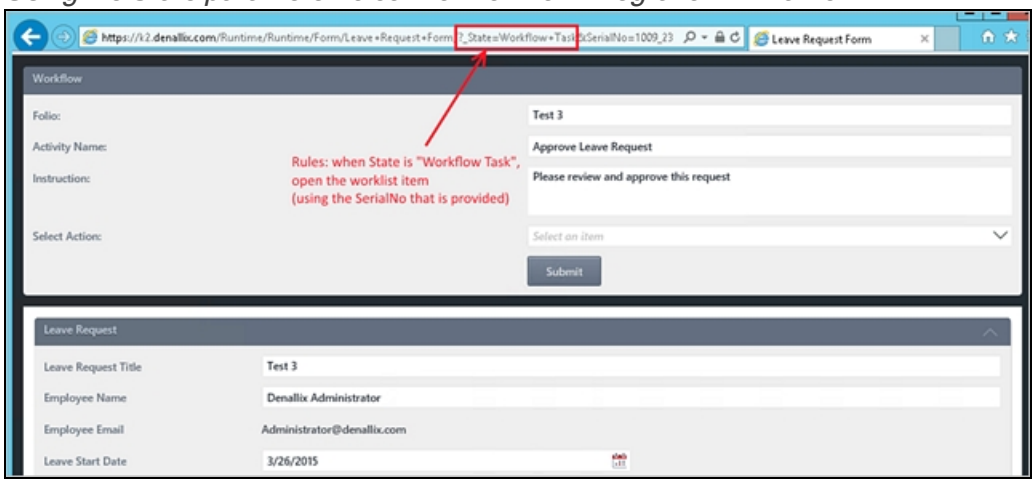
All Forms have a (Base State) with a collection of Rules. Any changes to the (Base State) Rules are automatically propagated to the other States as well. Subsequent States can their have their own Rules and behaviors applied on top of the Base State's functionality.

The Base State Rules are applied to all other states



States are often used for workflow integration because you can use the same Form to both start a workflow as well as during a workflow (e.g. an approval task). In other words, the form should behave differently depending on where it is used: to start the workflow or to complete a task during the workflow.

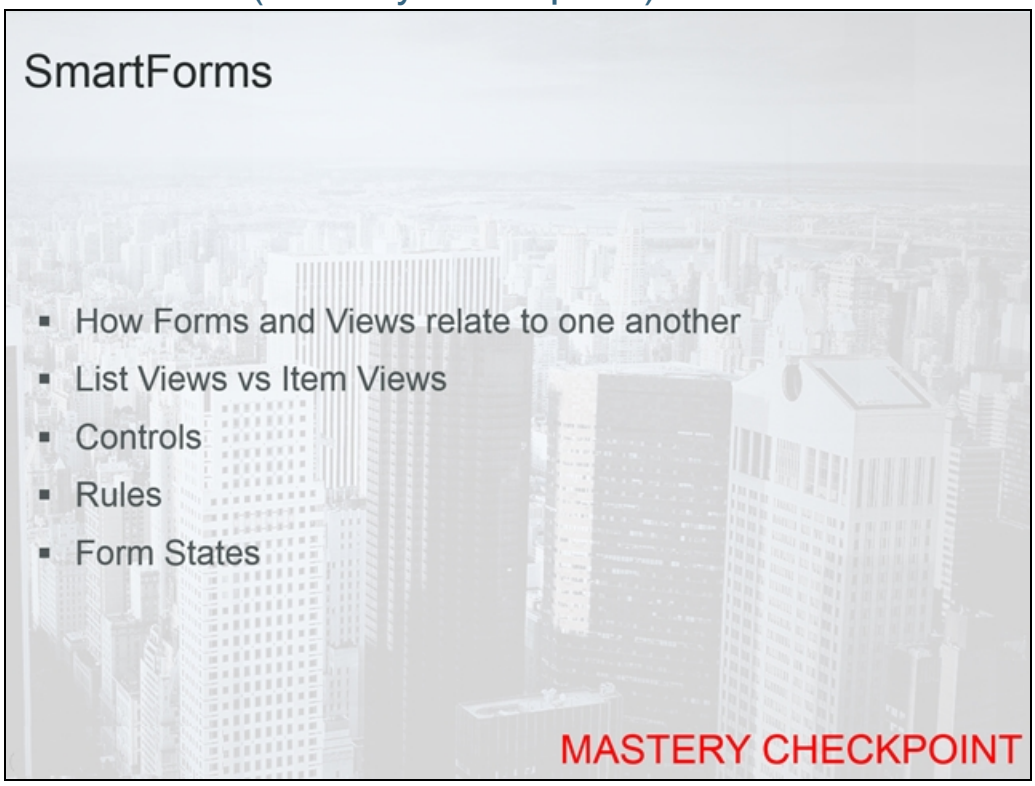
Using the State parameter to control workflow integration with a Form



Summary

- Form States are used to control the behavior of the same Form when it is being used in different contexts
- States are often used for Workflow integration
- States are passed in as query string parameters
- The (Base State) is the “foundation” for all the states in the form. By default all the rules and actions defined in the base state will be executed in all other States, UNLESS a State specifically disables that Rule or Action

SmartForms (Mastery Checkpoint)



Video

This is a checkpoint for the information covered in Part 2 of this module, focusing on SmartForms. If you are attending a training session led by a K2 instructor, this is your opportunity to ask questions on any topics you might not yet understand.

These are the main concepts you should understand:

- How Forms and Views relate to one another
  - Forms contain one or more Views
  - Views are usually “bound” to a specific SmartObject
  - You can use other SmartObjects in a View as well
- List Views (multiple records) vs. Item Views (single record)
- Using Controls on a View/Form
- How to define Rules on a View or Form (e.g. doing something if a button is clicked)
- What Form States are

Knowledge-check questions

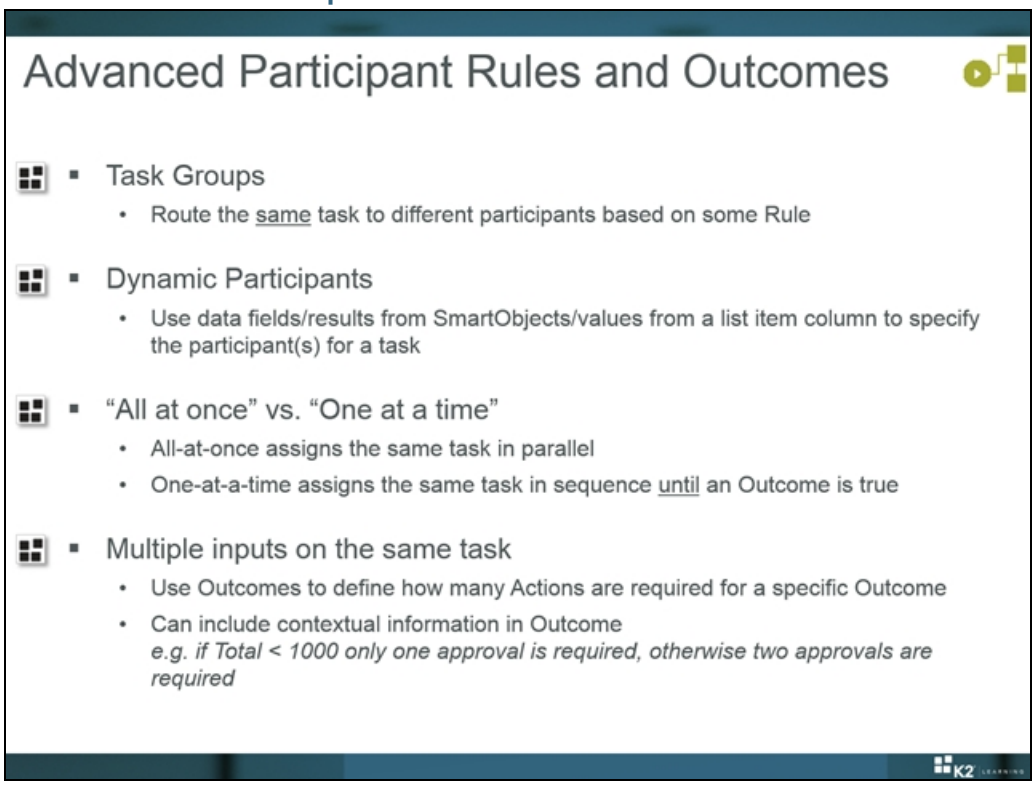
Q: True or False: A Form can contain only one View.
Reveal answer**A:** False, Forms can contain multiple Views (one or more).

Q: What is the difference between a List View and an Item View?
Reveal answer**A:** Use a List View to work with multiple records, use an item view to work with a single record.

Q:What are the three components of a Rule?
Reveal answer**A:** Event, Condition(s), Action(s)

Q:What is a typical use case for Form States?
Reveal answer**A:** To use the same Form for different stages of a workflow, to control the behavior of a Form depending on how it is being used.

Advanced Participant Rules and Outcomes



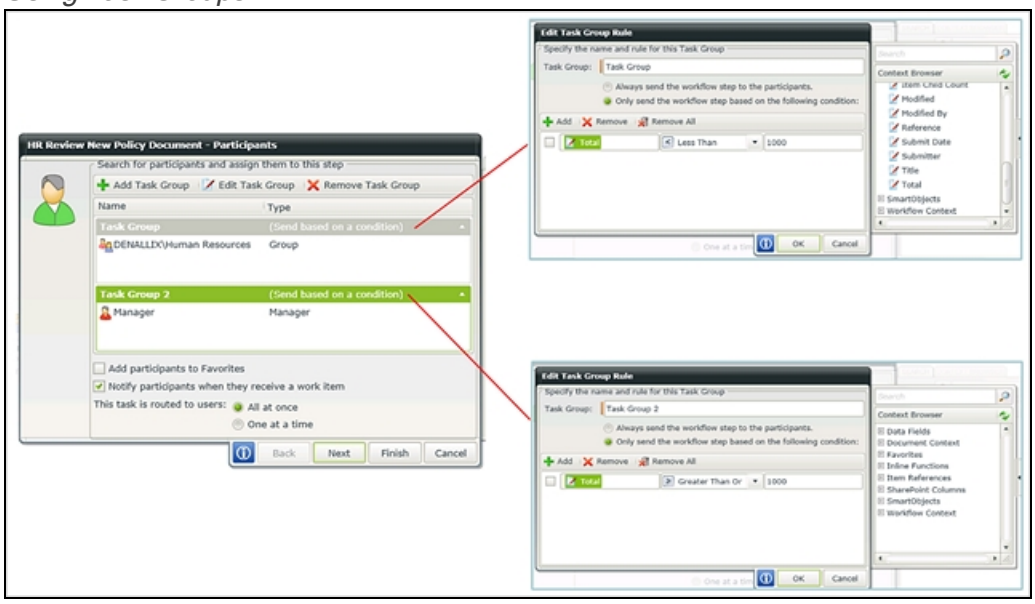
Video

You can configure more advanced Participant Task assignments as well as advanced Outcomes on Workflow tasks. Let's look at some of the available options.

Task Groups

Configuring Task Groups with underlying Rules allows you to specify if and when to send a task to a particular Task Group. For example, you might have a Form field called "Total" that contains a numerical value. If you wanted to send the approval task to a different group of users if this value is above a certain limit (for example, send the task to HR if the value is less than 1000 but to the employee's manager if the value is more than or equal to 1000, you can use Task Groups and Rules as shown below. This configuration allows you to use the exact same User Task step, but route the task differently based on some data value.

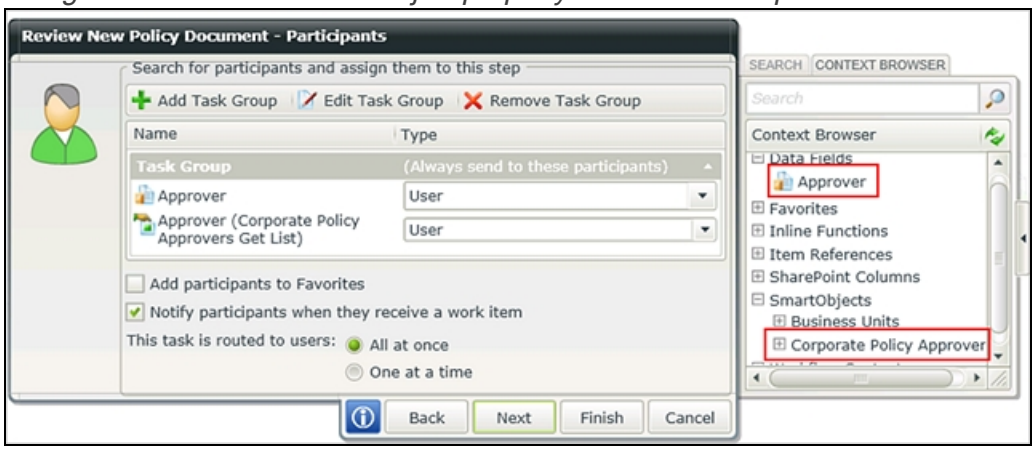
Using Task Groups



Dynamic Participants

You do not have to select Users or Groups when you configure the participants for a User Task. You can use dynamic values such as workflow data fields or a result from a SmartObject method to populate the value of the participant. Suppose you wanted to maintain a list of Corporate Policy approvers in a SharePoint list. All you need to do is create a SmartObject (Data) for that list in SharePoint, select the option to expose the SmartObject to workflows and then you can use this new SmartObject in the Participants for a workflow step. In the future, if you need to modify the person who approves Corporate Policies, you only need to modify the value in the SharePoint list, and the workflow will automatically start routing subsequent tasks to the new user(s).

Using a Data Field or a SmartObject property to set the Participants for a Task

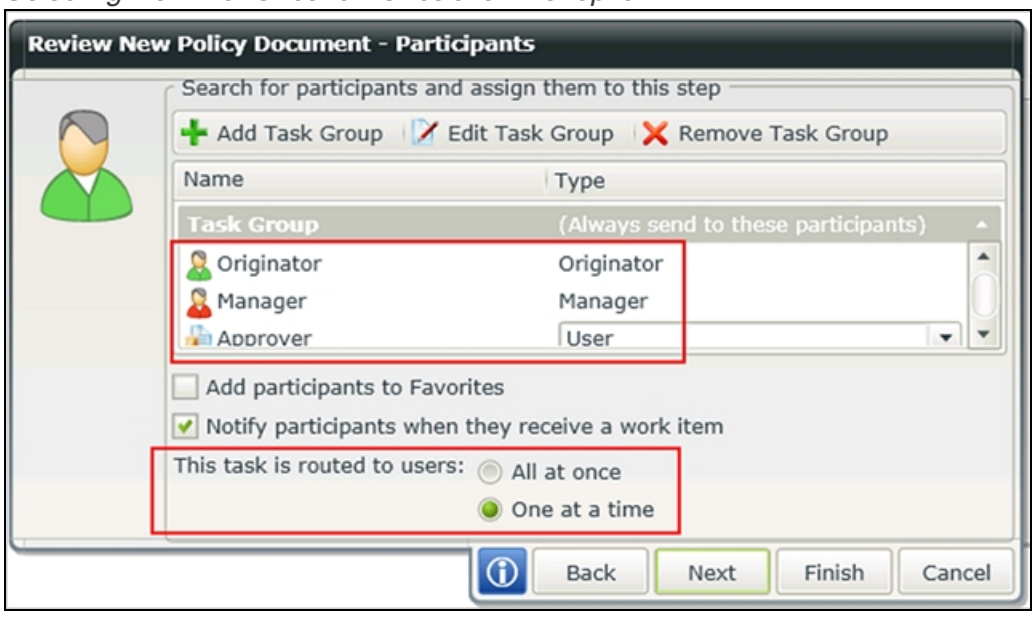


Serial or Parallel Task Assignment (All at Once vs. One at a Time)

Suppose you have a User Task step where a group of users are set as the participants for the task. When you select the "All at Once" option, the task is assigned to all the users in that group simultaneously. As users complete the task, K2 will check the outcome and if the outcome is not successful, K2 waits for the next user's decision, and so on. The key here is that the task is assigned to all of the users at the same time and anyone can open the task at any time. You can think of this as a "parallel" task assignment where the task is assigned to multiple users at the same time.

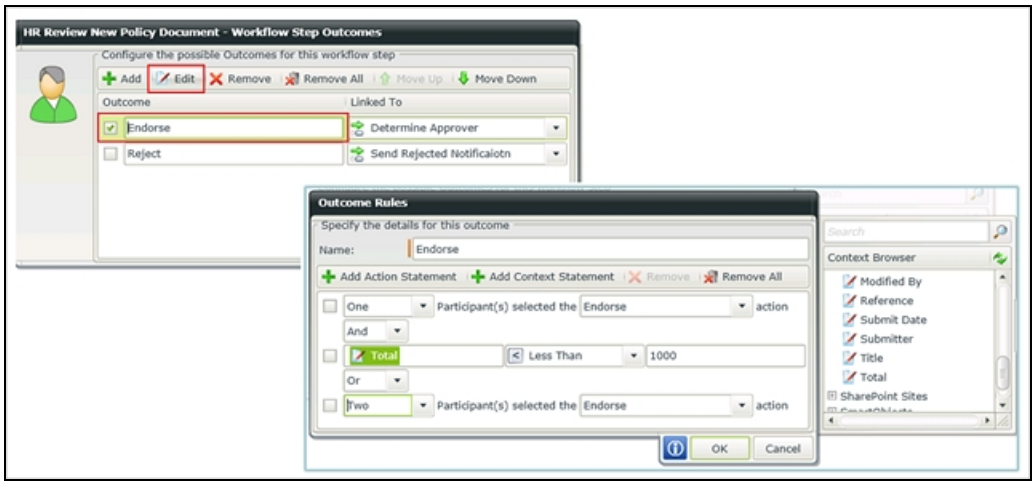
If you select the "One at a Time" option, K2 will assign the task to the users as they are listed in the Task Group or returned when the group membership is resolved. This means that the task is assigned to the first user first. When that user completes the task, K2 checks the outcome. If the outcome is true, K2 completes the task and continues with the workflow. If the Outcome is not true, K2 assigns the task to the next user in the group and waits for their response, and so on until at least one Outcome is satisfied. You can think of this style of task assignment as "serial" where the task is assigned to a number of users but in sequence, one at a time.

Selecting the "All at Once" or "Once at a time" option



Advanced Outcomes

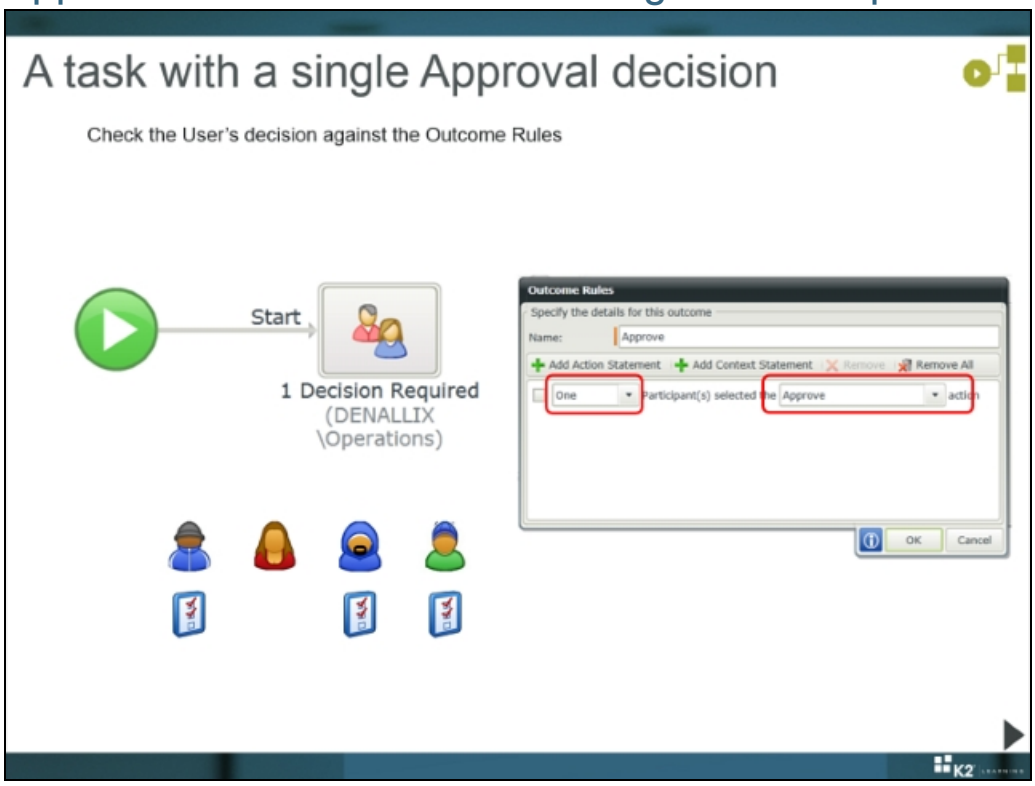
When configuring outcomes, you can decide how many participants need to provide input and you can also evaluate a dynamic data value at the same time. Here is an example: suppose that if the total value of a proposal is less than 1000, you only need one user to endorse the request. However, if the value is more than 1000, you require two users to endorse the request. You can use an outcome like the example below to configure this exact behavior. The value on either side of the expression could come from anywhere: it could be a static value, a data field value or even a value returned by an Inline Function or a SmartObject property.



Summary

- Use Task Groups with group rules to route the same task differently depending on conditions
- Use Context fields or SmartObject methods to dynamically specify the Participants for a User Task
- Parallel vs serial task allocation (“All at once” vs “One at a time”)
- Using Outcomes to specify multiple inputs on the same task
- Advanced Outcomes that evaluate data as well as user input

Approval Decisions and Slots: Single vs. Multiple decisions



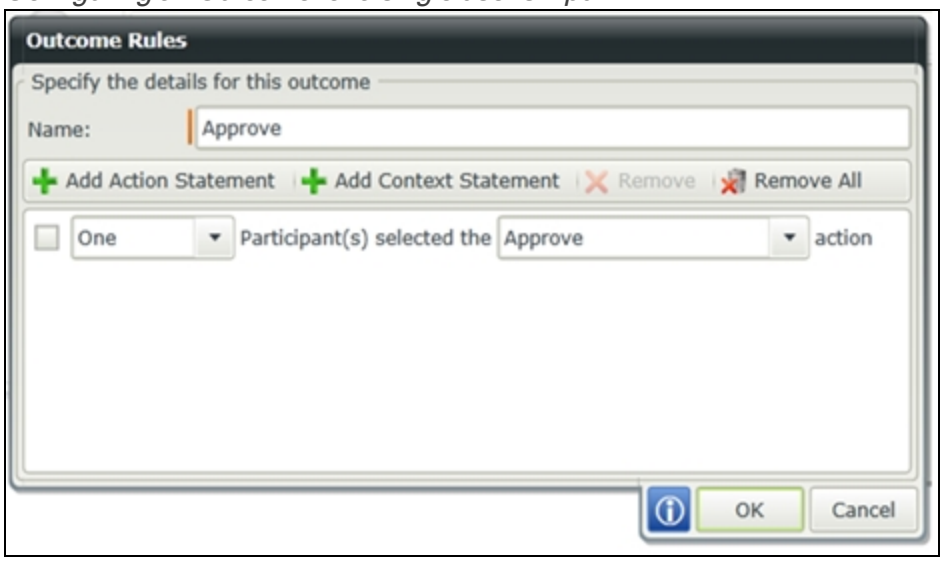
Video Video

You can configure the Outcomes for a User Task so that only one user in a Group needs to action and complete the task, or multiple users need to action and complete the task. To do so, edit the Outcomes of the User Task in question.

Single Decision

In this scenario a task is assigned to a group of people. As soon as one person has approved the task, K2 will evaluate the Outcome Rules to see if they have been met. In a single decision scenario, the Outcome Rule might specify that 'One' person must 'Approve' the task. So in this case, the Rule has been met and the remaining tasks assigned to the other group members will expire and the workflow will move along its configured path based on the successful outcome.

Configuring an Outcome for a single user's input

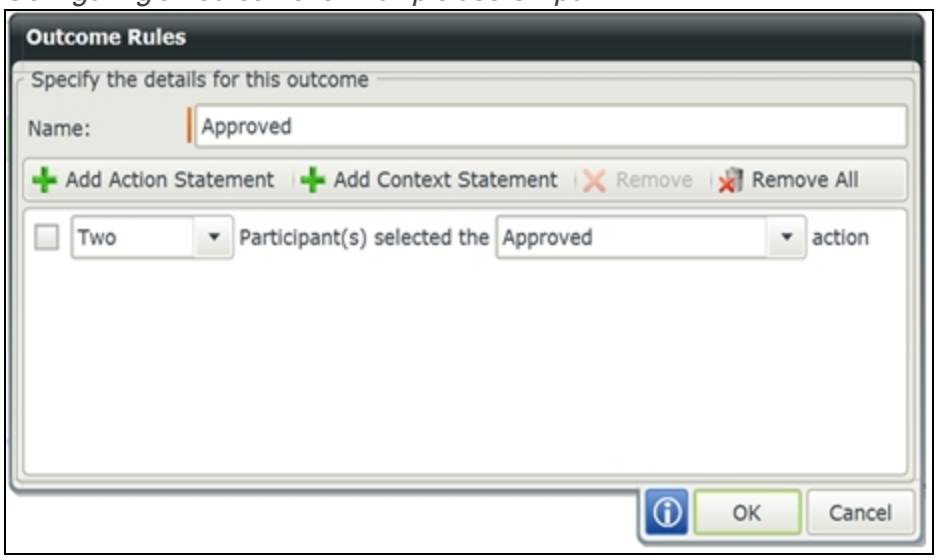


Multiple Decisions

In this scenario, the Outcome Rule has been configured so that at least two people from the assigned group must approve the task. After the first person approves the task, K2 evaluates the Outcome Rules as it did in the previous

example. Since the Outcome Rules state that two people must approve the task, the remaining assigned tasks are not expired because we still need one more person to approve the task. The available slots will remain active until a second approval is made. Once the second approval is made and assuming that the outcome has been met (the second user selected "Approve", in other words), the remaining assigned tasks will be expired and the workflow moved along its configured path.

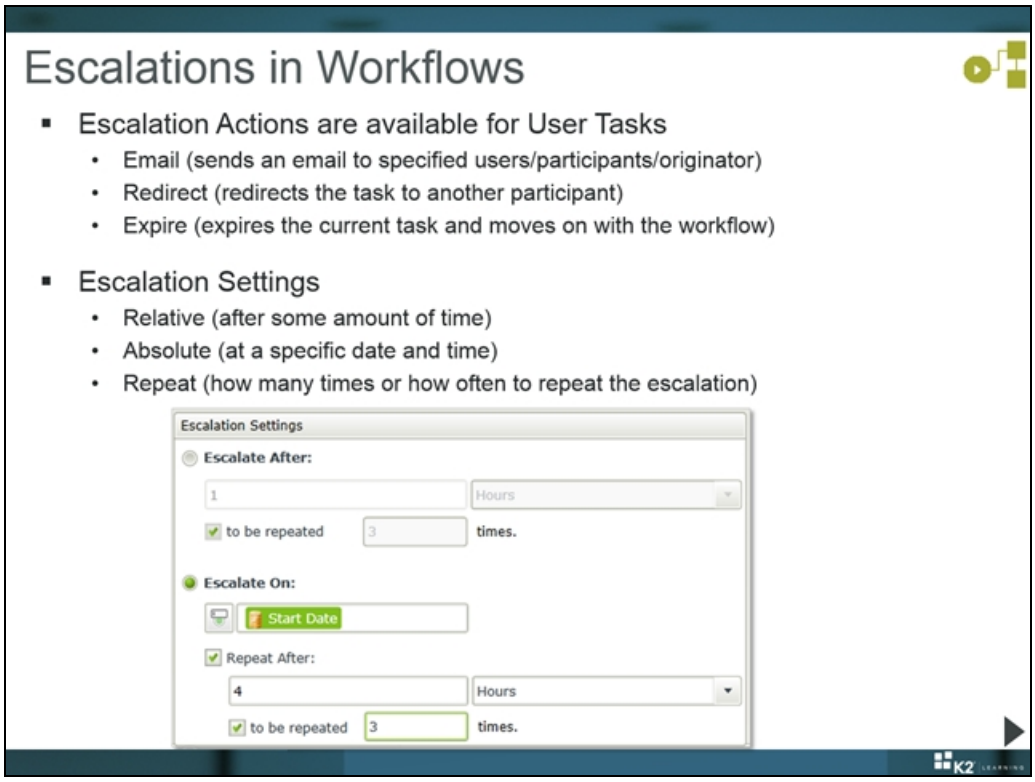
Configuring an outcome for multiple users input



Summary

- Use Outcome Rules to configure how many users need to provide input on a User Task
- With one slot, as soon as one user opens the task it disappears from the other user's task lists
- Users can "release" the task back to the group
- With multiple decisions, the task is not completed and the workflow will not continue until one of the possible outcomes is True

Escalations in Workflows

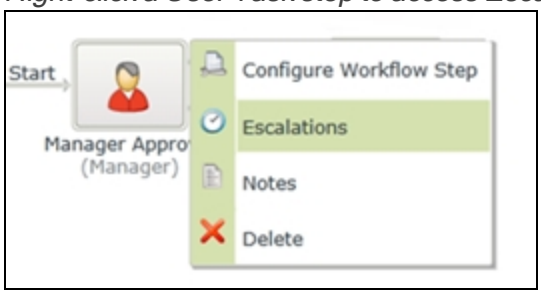


Video

In a workflow, Escalations can be configured for User Tasks to fire when a task has gone beyond it's expected completion time. Escalations can be as simple as an email reminder notice or they can be more complex, such as redirecting the task to another user or expiring the task and moving on to the next workflow step.

Note
Escalations cannot be configured for System Tasks, they are only available for User Tasks.

Right-click a User Task step to access Escalations

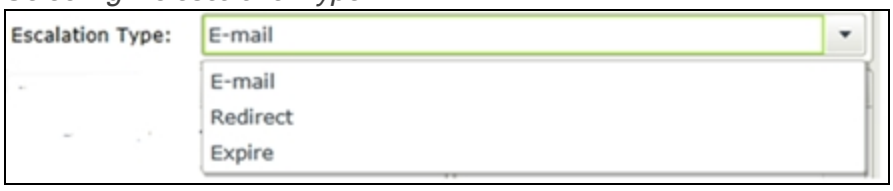


Escalation Types

There are three types of escalations:

- Email: email escalations will send an email to specified user, task participants or the workflow originator
- Redirect: sends the User Task to another participant
- Expire: expiring a task means that the task will be expired (considered complete) and the workflow will then move on to the next step or event

Selecting the escalation type



Escalation Settings

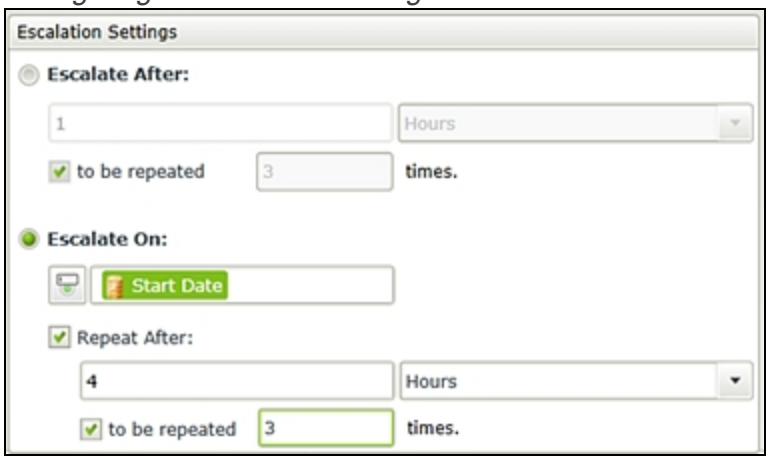
Escalations can be configured to start based on the following:

- Relative: starts the escalation after a specific amount of time. For example, you might configure the escalation to fire 3 days after the task was assigned.
- Absolute: escalations start on a specific date and time (this could be a calculated date, such as the last day of the current month).
- Repeat: how many times and how often. For example, your escalation might repeat every 3 days, 3 times.

You can use dynamic values from the Context Browser in your Escalation settings. For example, you might define the escalation intervals in an external list that's referenced by a SmartObject. Then, if you need to update the escalation interval, you simply change the value in the list without affecting the workflow configuration or having to redeploy the workflow.

Escalations begin when the User Task is assigned. If the User Task is completed within the expected time frame, the escalation will expire. If the User Task is not completed, then the escalation will fire with the type and interval configured.

Configuring the escalation settings

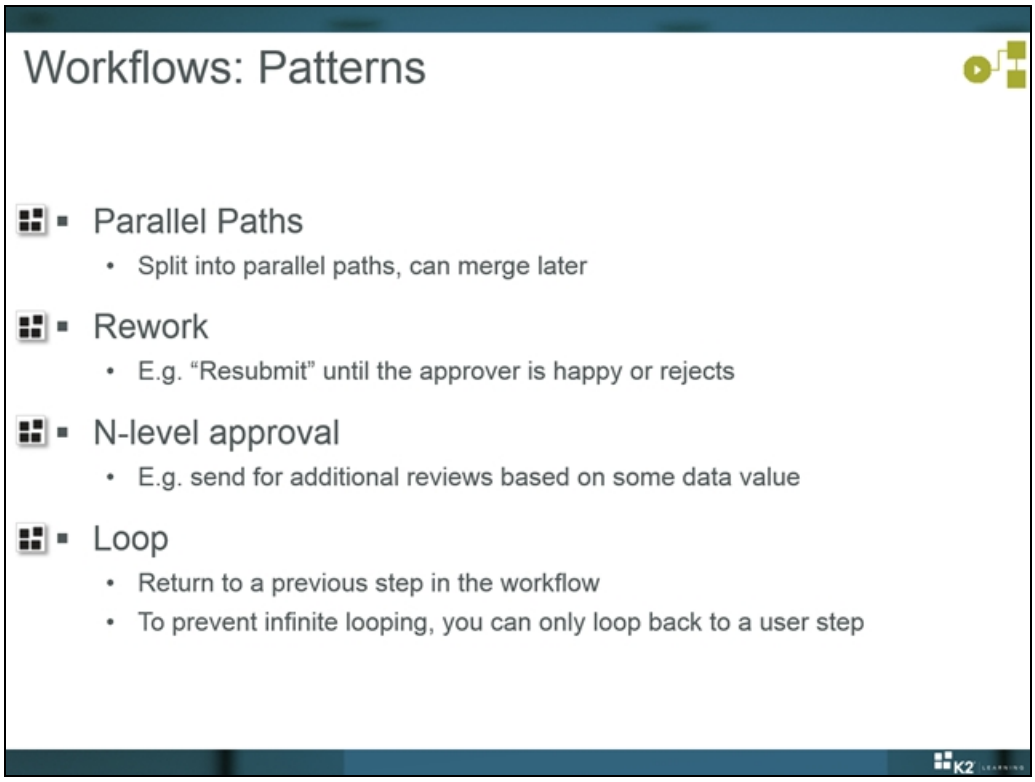


Tip
Be sure to consider your overall design when working with escalations. You can easily overflow a manager's inbox with escalation notifications from multiple workflows. Keep in mind who you are directing the escalation to and how often. One tip is to send an escalation notification back to the originator so they can follow up and get their workflow moving again.

Summary

- Escalations are initiated when the workflow step starts and are expired if the step completes before the escalation is reached.
- Be careful not to over-use escalations.
- It is possible to define the interval for the escalation statically or dynamically.
- In the browser-based workflow designer, you can only define escalations for User Steps in a workflow, not for server steps.

Workflow Patterns



Video

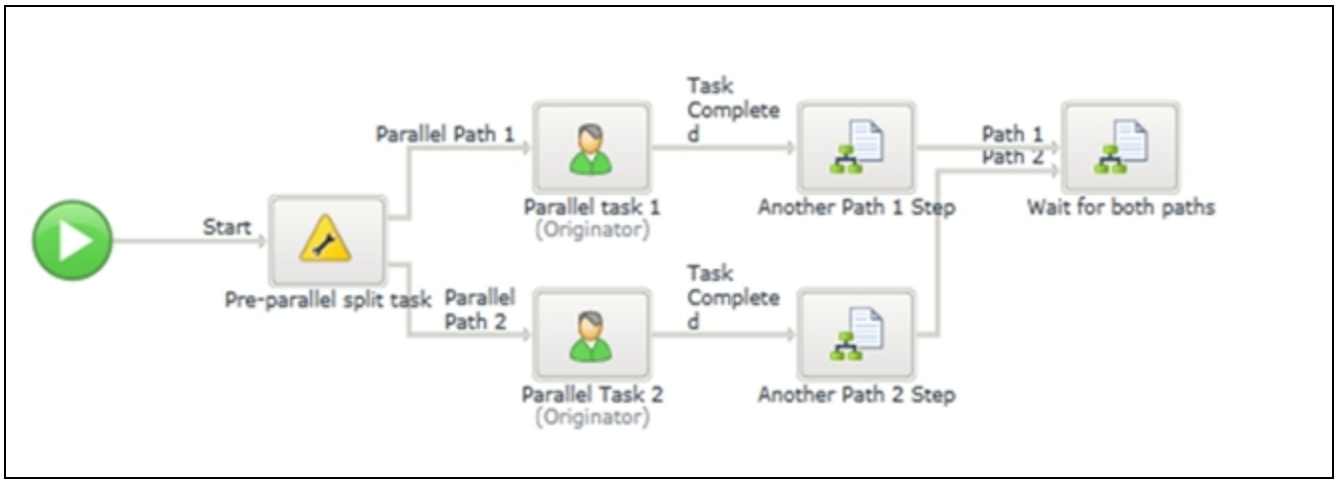
It is possible to build several types of workflow patterns in K2 Designer. This topic discusses some typical patterns and how you may implement them.

Parallel Paths

It may be efficient for the process to split into multiple parallel paths and join again at a later stage. For example, it may be efficient to deliver tasks to two different departments simultaneously, so that neither department has to wait for the other to complete their work (assuming of course that the departments do not depend on each other’s work).

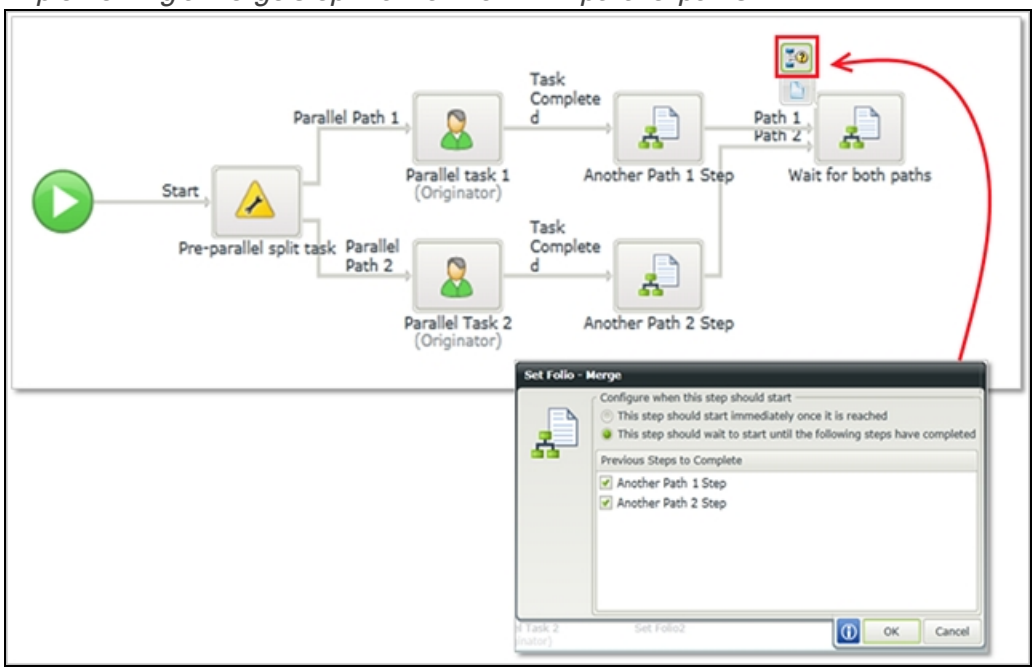
K2 will always follow all possible outcomes in a process, so if two outcomes are both true, the process will split into a parallel execution model.

Consider the process design example below. In this process, the workflow will split into two parallel paths after the “Pre-parallel split task” completes. The subsequent parallel User tasks will be delivered to the respective destination users simultaneously, and either can complete their task at any time.



The key to making parallel execution work for a merging step is setting a property on the task where the two paths need to merge. Consider the final “Wait for both paths” task in the diagram above – we would not want the task to be executed twice (one when Path 1 completes and again when Path 2 completes). Instead, we want to wait for both Path 1 and Path 2 to complete before allowing the final step to execute. To achieve this, configure the “Merge” setting on the final task to wait for both paths to complete

Implementing a Merge step in a workflow with parallel paths

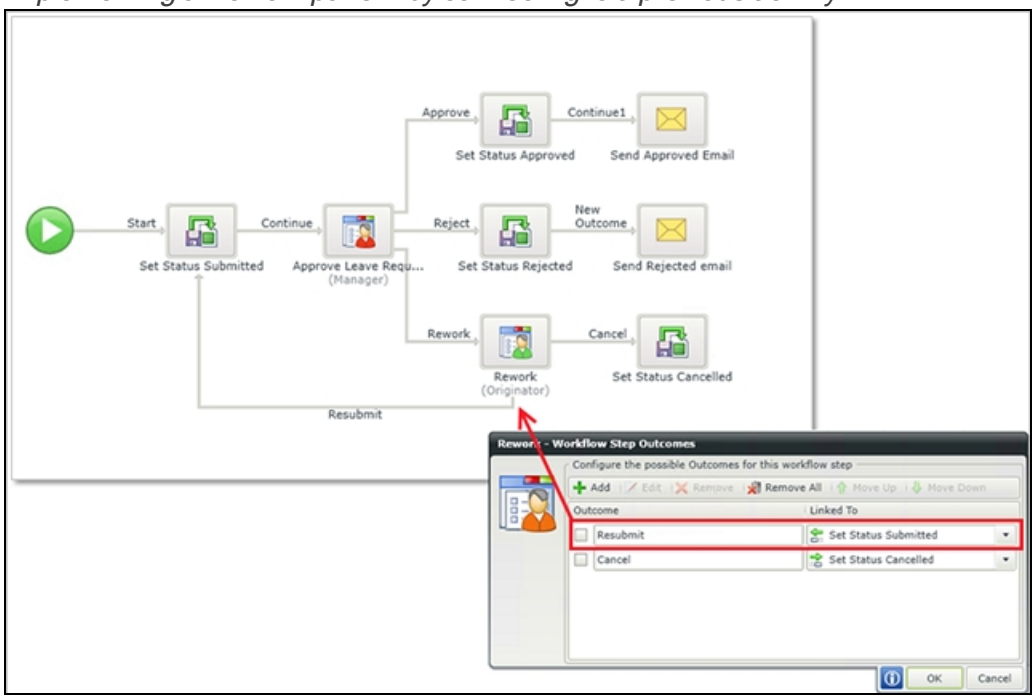


Rework

A very common design pattern is the rework-loop pattern. In this pattern, a step of the workflow may repeat any number of times until some condition is met and the process continues. It is most often used in an approval scenario where the approver can send the request back multiple times until they are happy with the request and select a different action to continue with the process.

The figure below is a sample of this scenario. In this example, the Approval step has three possible outcomes: Approve, Reject or Rework. If the approver selects Rework, the task is sent back to the originator of the process so that they can adjust the request and submit it for approval again. Because this loop uses the same steps in the workflow, this rework loop could execute any number of times, until the approver is satisfied with the request and approves or rejects it.

Implementing a Rework pattern by connecting to a previous activity



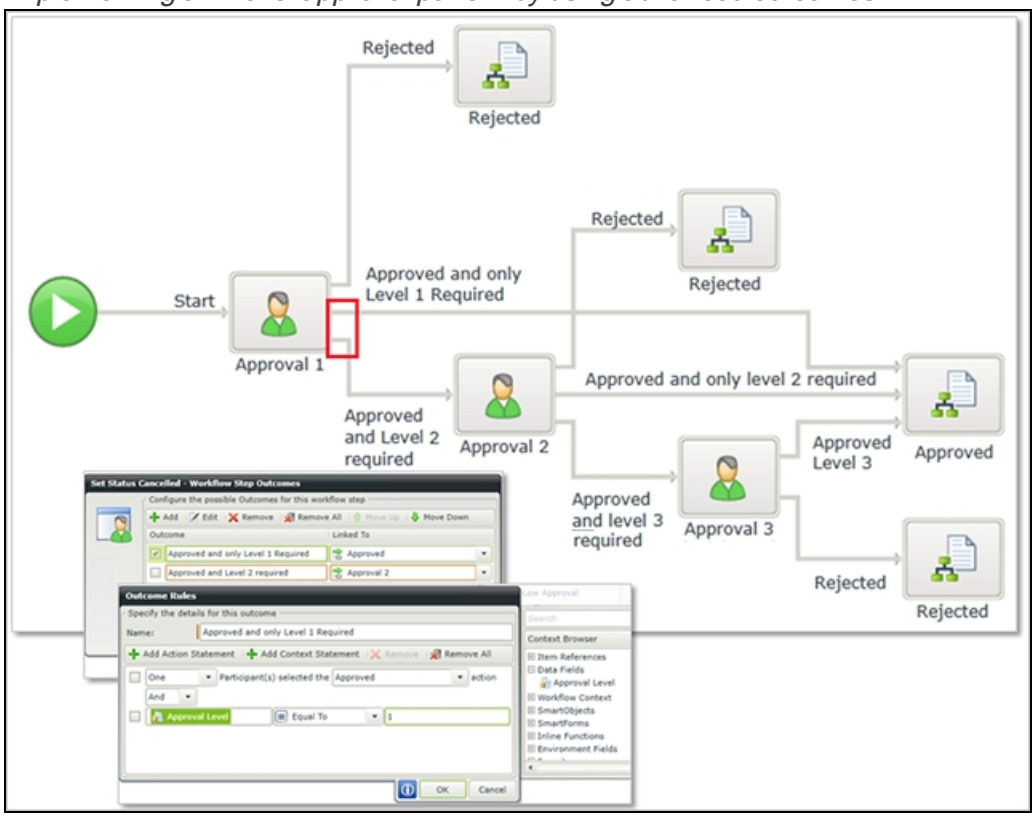
N-level Approval

Often, a process may have an unknown number of approval steps. For example, a generic document approval process may require anywhere from one to five levels of approval, depending on the nature of the document.

The N-Level approval pattern is a useful pattern when there are a known maximum number of approvals, and any particular process instance may require several levels of approval which will be determined at execution time.

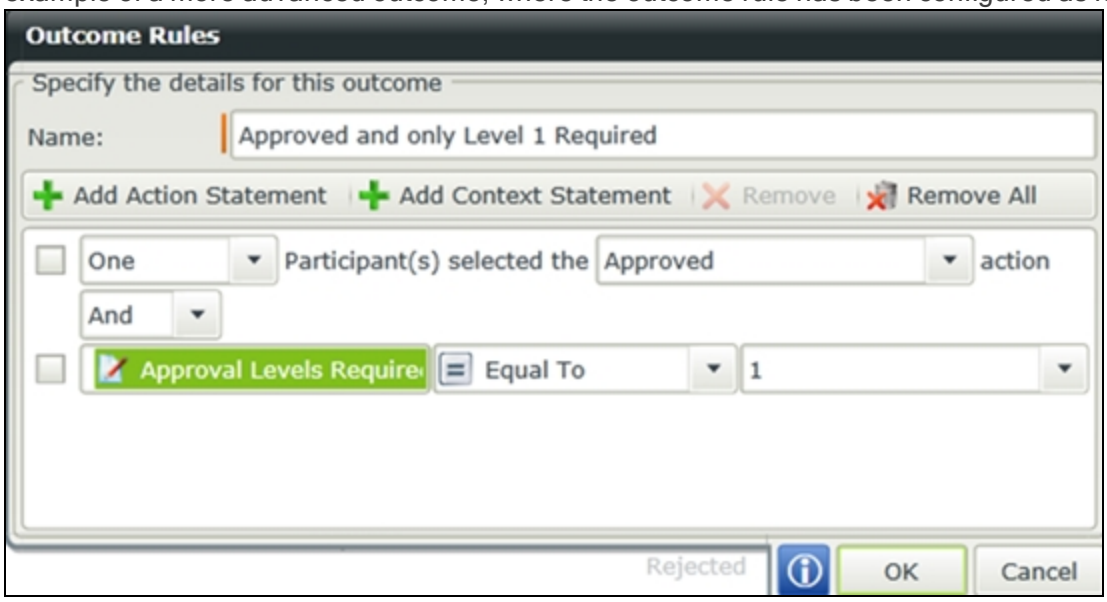
The example below illustrates a process where the originator will decide how many levels of approval are required. Each outcome from the approval steps will evaluate the required number of approvals, and if there are no more approvals required the process will skip the remaining approvals and go to the final “Approved” step.

Implementing a N-level approval pattern by using advanced outcomes



Consider “Approval 1” in this diagram. The Approver has two possible Actions: Approve and Reject. The “Rejected” outcome is straightforward – if the approver selects “Reject”, go to the “Rejected” task.

The “Approve” action, however, is associated with two possible outcomes: “Approved and only Level 1 required”, and “Approved and Level 2 required”. The approver still selects “Approve” as the action, but each outcome also evaluates the “Number of Approvals Required” data field which was specified by the user that submitted the request. This is an example of a more advanced outcome, where the outcome rule has been configured as follows:

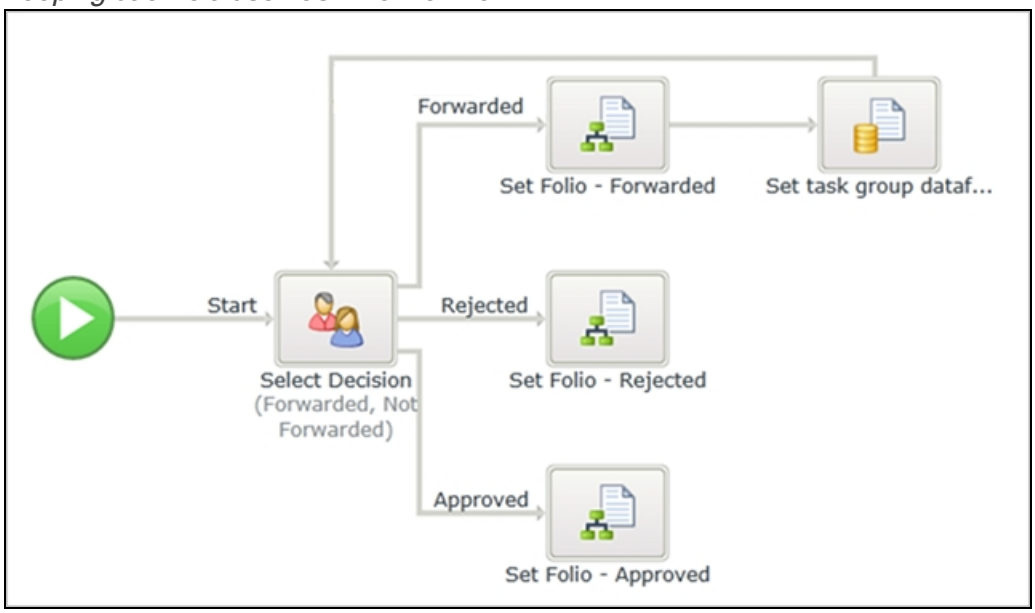


Loop

Similar to the rework process described above, the looping process can repeat a particular step in a workflow as many times as necessary. An example may be a process where an unknown and potentially large number of approvals may be required – in this case, the process designer could repeat the approval step until all the required approvals have been gathered, and then use outcome rules to continue executing the process normally.

The figure below is an example of a process where a user may select one of three possible decisions: Approve, Reject or Forward to Another User. If the user selected the “Forward to another user” option and specifies an alternative user, the process will perform some tasks and then loop back to the approval step, but this time, the approval step is assigned to the other user. That user, in turn, may select any of the possible outcomes, including forwarding the task, effectively repeating the cycle again.

Looping back to a user task in a workflow

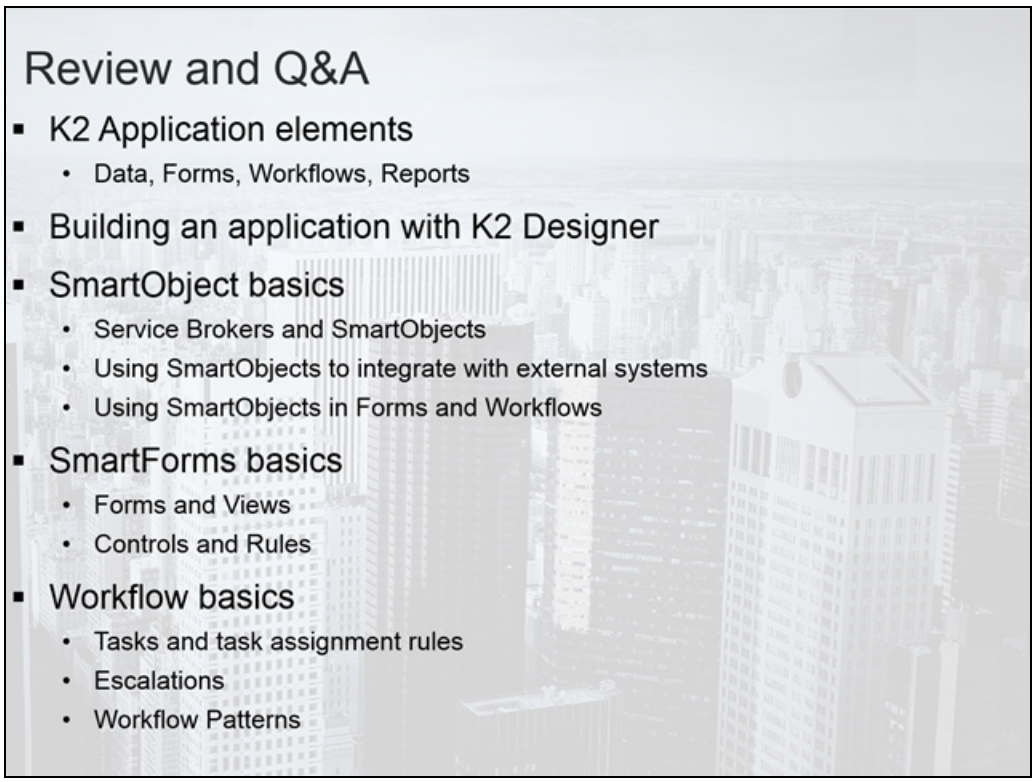


Caution
Looping processes have a real possibility of creating an infinite loop, which will negatively affect the performance and database growth of a K2 environment. You should NEVER use the looping model with server-only tasks - there should always be a user task as part of the loop or some condition that will cause the loop to exit. User tasks effectively “pause” the loop, because K2 is waiting for a user to complete the task before the process will continue. If you used Server Tasks only, there is no “Pause” and the process will loop constantly until the loop is exited, usually with an Outcome.

Summary

- Parallel Paths
  - Split into parallel paths (steps run concurrently)
  - Can merge the paths back together again later with the start condition of the step where the paths merge (wait for both preceding paths to complete)
- Rework
  - E.g. “Resubmit” until the approver is happy or rejects
  - Link an outcome back to a previous step in the workflow
- N-level approval
  - E.g. send for additional reviews based on some data value
  - Implement the rule in the outcomes to determine whether to go for 2nd approval or not
- Loop
  - Return to a previous step in the workflow

Review and Q&A



Video

This topic is just a summary of the information covered in this module: Introduction to K2 Applications with K2 Designer. If you are attending a training session led by a K2 instructor, this is your opportunity to ask questions or, if time permits, discuss your own plans for using K2 in your own environment.

In summary, here is what we covered in this training module:

- How to build simple applications with K2 Designer
- The basics of using SmartObjects to integrate with other systems
- SmartObject architecture
- The basics of using SmartForms as the user interfaces for applications
- The basics of workflows, such as User Tasks vs System Tasks and how to define Actions, Outcomes and Escalations

Knowledge-check questions

Q: How would the sample Leave Request Approval application look in your organization? What would be different in the workflow? Or the Forms?
A: (discussion question)

Q: What data sources in your own organization might you use if you were to build a similar Leave Request Approval application in your organization? Where are those data sources?
A: (discussion question)

Q: What is the difference between Data, Forms and Workflows in a K2 application? What is each component responsible for?
A: (discussion question)

Q: How do you add an escalation to a workflow? And why would you want to?
Reveal answer
A:Right-click a user task. Escalations are actions that K2 will perform automatically when the task is not completed within or by a specific time, e.g. send a reminder to action the task.

Business Analysis for K2 Applications

This learning module **200.BEL Business Analysis for K2 Applications** describes how to qualify, analyze, discover, and document business requirements for K2 Applications.

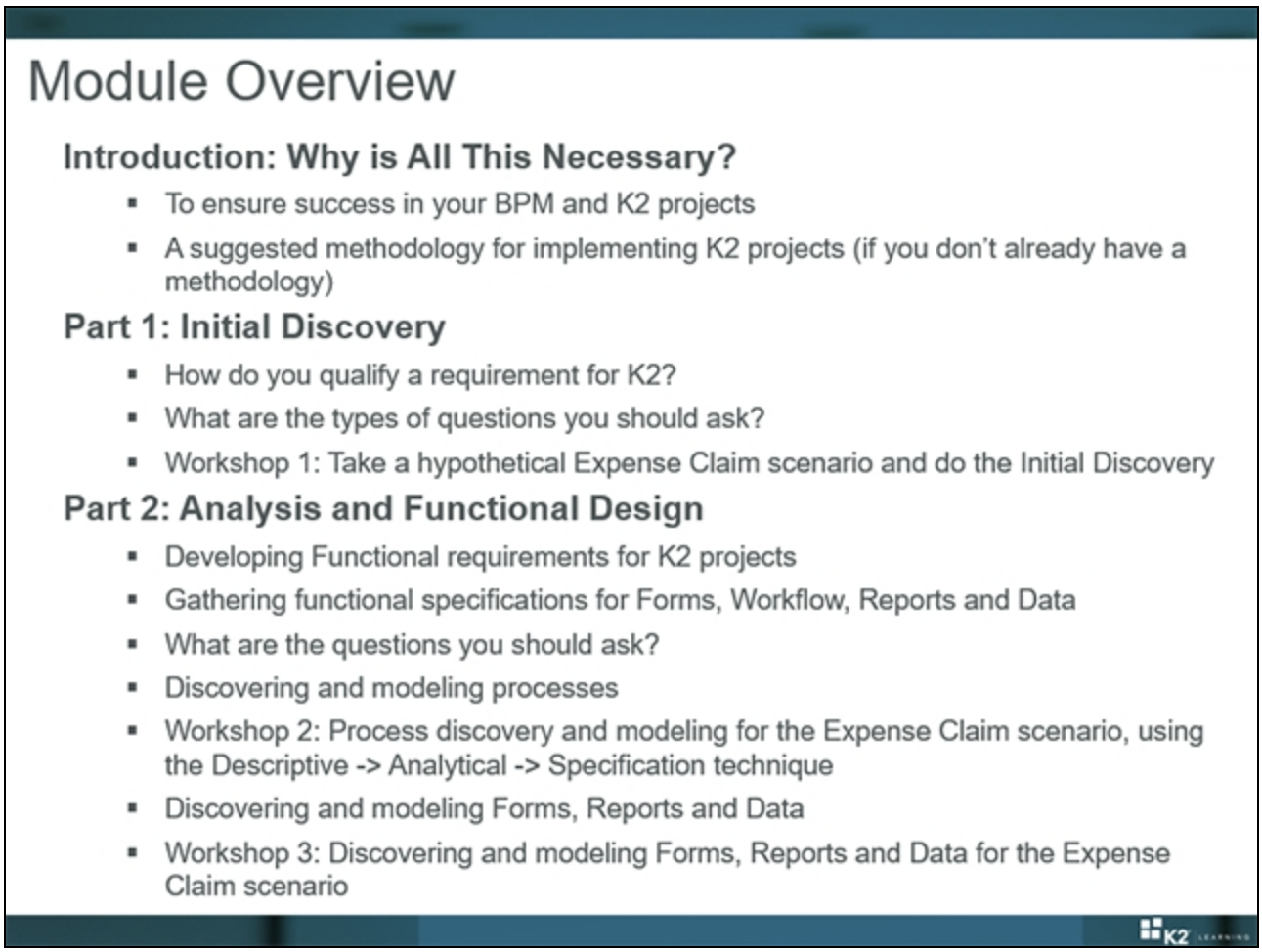
This module is intended for Business Analysts who will be qualifying business requirements against K2 and then gathering and documenting the business requirements for the K2 application.

We expect people in these roles to have basic K2 knowledge and familiarity with Business Analysis principles like requirements gathering and requirements discovery.

In this module we will be talking about the typical life cycle of K2 projects, a suggested methodology for implementing requirements in K2. We will discuss how to qualify requirements through the initial discovery and analysis for K2 projects and how to discover and model the components of K2 applications: Forms, Workflows, Reports, and Data.

Note that this module is not intended to cover the technical design, implementation, or development of applications with K2. Those tasks are typically performed by more technical roles like designers and developers and is covered in other K2 training courses. This module is focused on starting your K2 projects correctly with the right approach and the right requirements.

Module Overview



This module is divided into 3 main sections.

In the Introduction we will quickly cover why all this analysis is necessary and look at a suggested methodology for implementing K2 solutions.

In Part 1 we will look deeper into he Initial Discovery phase of K2 solutions: how to qualify a requirement for K2, what types of questions you need to ask to establish the business case for the applications. This will be followed by a workshop where we will take a hypothetical expense claim approval requirements and put it through the Initial Discovery steps, learning what questions to ask and why it is important to ask them.

In Part 2 we will look into the Analysis and Functional design stages of a K2 project life cycle. Here you will learn how to gather requirements and document functional specifications for K2 projects by breaking them down into their components: Workflow, Forms, Reports, and Data.

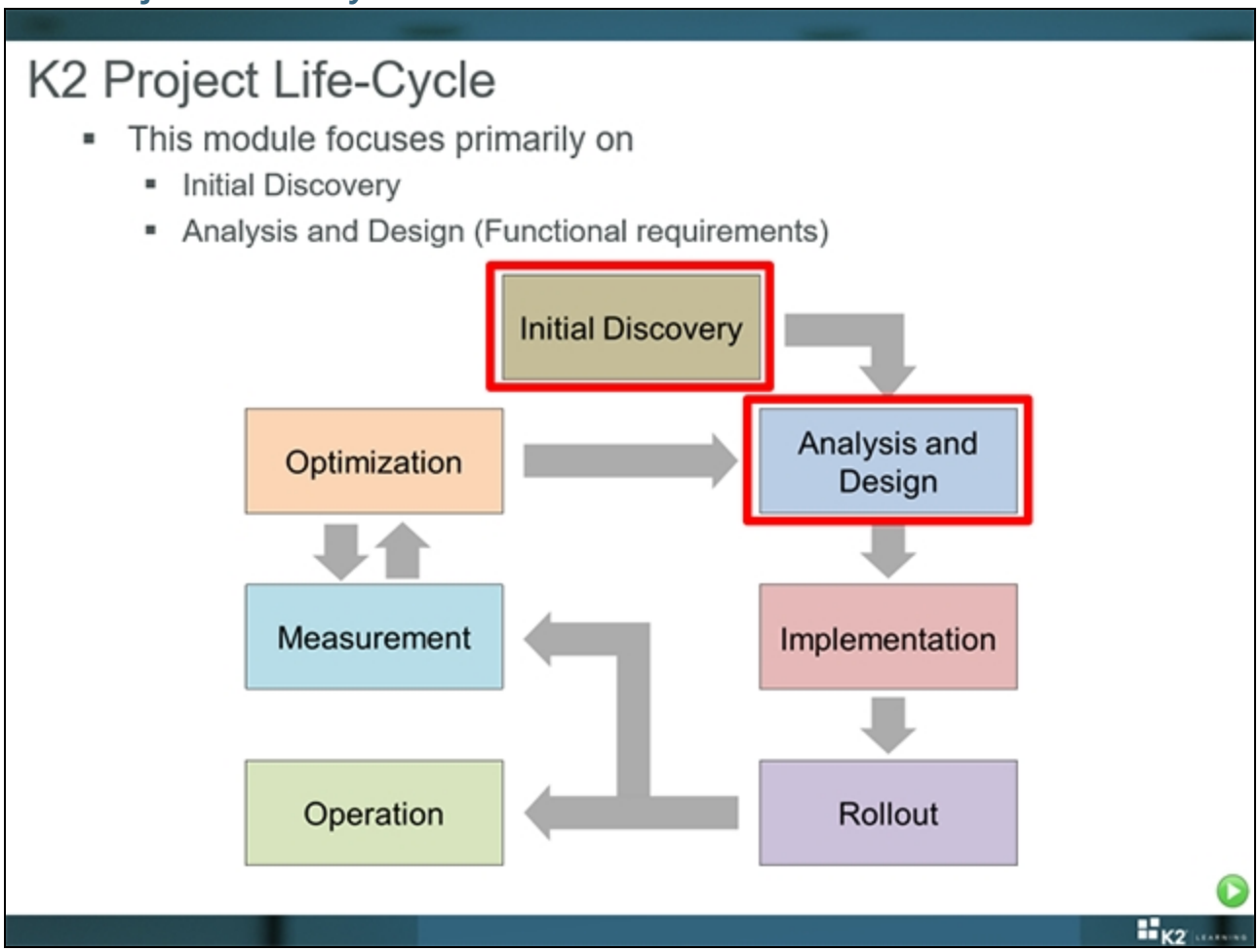
We will focus more on discovering and modeling workflows since this is often the central part of K2 applications. Once we've described how to model workflows, we will do a workshop where we will take the hypothetical expense claim approval workflows and run through a typical process discovery session using the Descriptive > Analytical > Modeling approach.

After the process discovery workshop, we will look at how you can model and discover the other typical components of K2 applications: Forms, Reports, and Data, followed by a workshop session where we will take these components of the expense claim approval scenario and discover and model them.

In Part 3 the module wraps up with an overview of common best practices when it comes to thinking about the functional requirements in K2 and how best to leverage the power of the platform.

Note
The workshop exercises in this learning module use several resources, for example template documents and workshop guides. If you have not done so already, please download the following file from K2's help site: http://help.k2.com/files/10701.
Once the zip file is downloaded, please extract the contents of the zip file to your computer. (Alternatively, you can also extract the zip file to the virtual K2 environment provisioned for you for this training course.)

K2 Project Life-Cycle

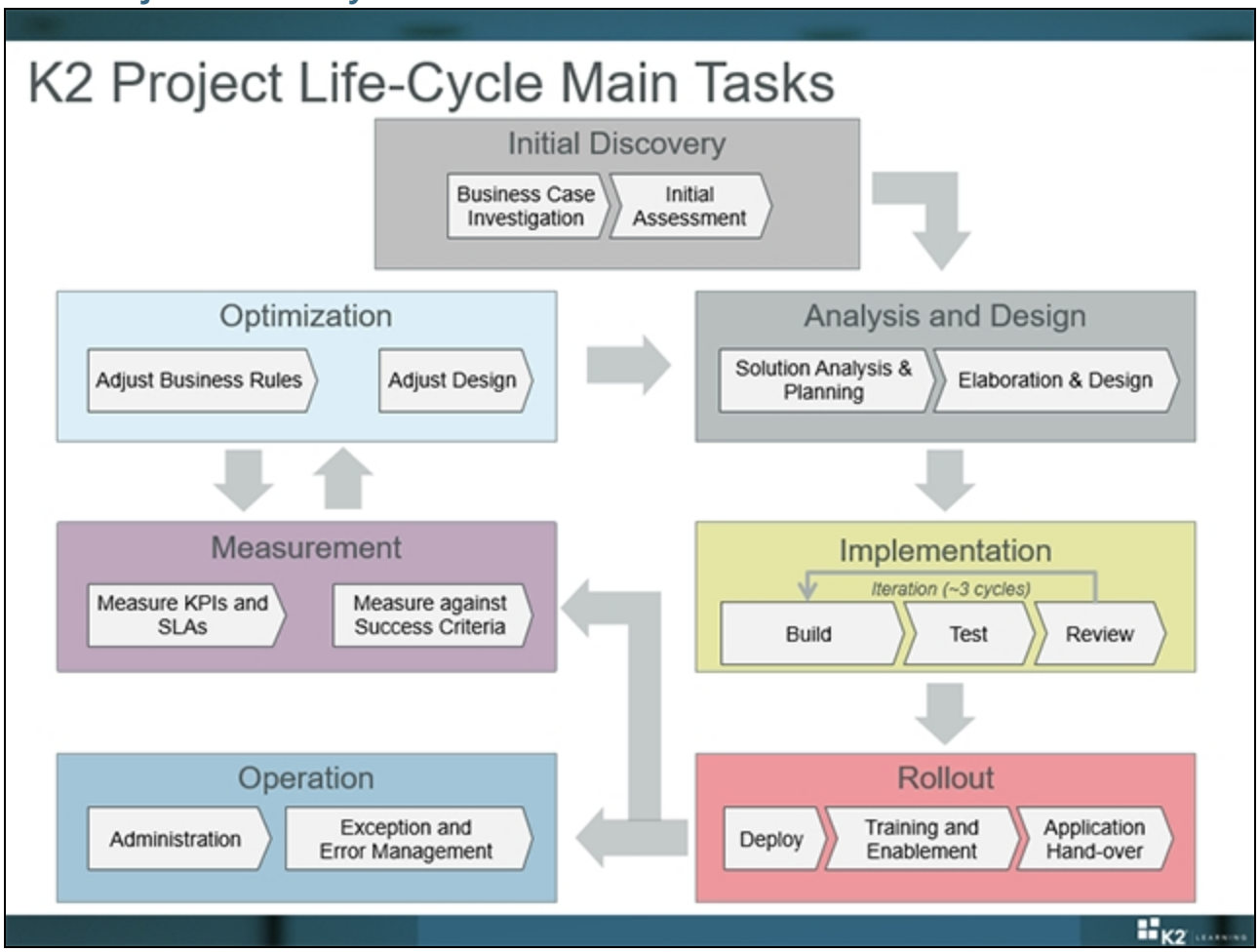


This diagram illustrates the typical life cycle of a BPM project. We will not cover the entire project life-cycle in detail in this module. This particular learning module is mostly concerned with the “Visioning and Scope” and “Design” phases of the BPM life cycle. The other phases are covered in other learning modules, which may be specific depending on the organization’s intended use of K2 (e.g. whether you intend to use SmartForms vs. custom web pages, or code-based development vs. no-code development).

The two stages we will focus on in this module are:

1. Initial Discovery – this is usually a one-time event and the entry point for a project. Here we establish the purpose and business case of the BPM project.
2. Analysis and Design – this is where the requirements and specifications are gathered and documented. This module stops once the functional requirements have been documented. After that you would typically hand the requirements to a designer or developer to implement.

K2 Project Life-Cycle Main Tasks



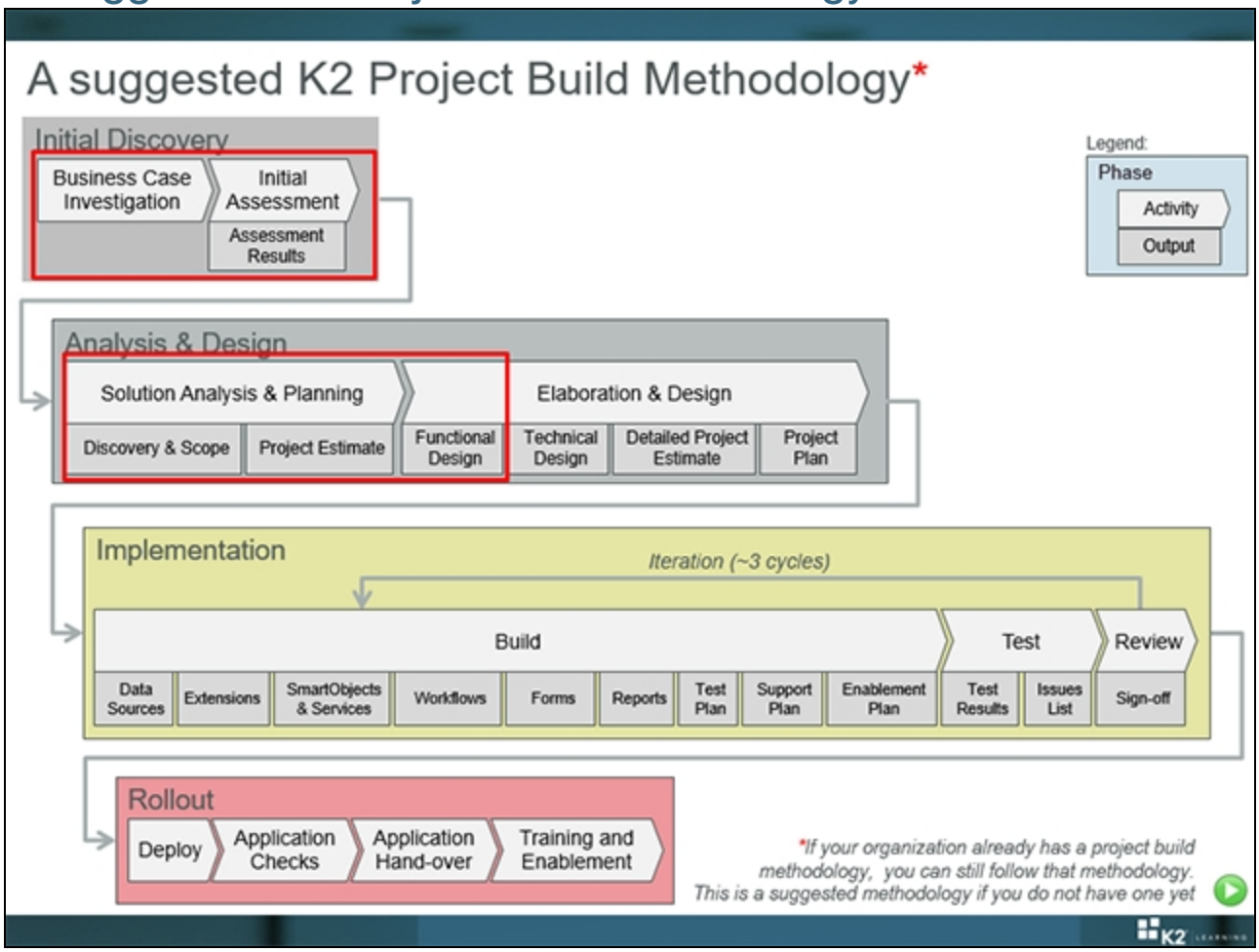
Let's break down a typical K2 BPM project life-cycle into the various phases we usually see.

We can break the BPM life cycle for a project into the following stages:

1. Initial Discovery – this is usually a one-time event and the entry point for a project. Here we establish the purpose of the BPM project.
2. Analysis and Design – this is where the business requirements are discovered and documented and the functional requirements documentation is generated. Once these documents are available, technical roles will convert he functional requirements into technical specifications., which are then fed into the Implementation phase.
3. Implementation – this is when the project is implement by the developers/designers. This may include prototyping, iterative development cycles of the actual solution components, testing and QA and deployment planning. Most important to note is that the development process is usually iterative so ensure that the solution components are meeting the requirements.
4. Rollout – this is where the application is deployed to the organizations. This usually includes tasks to deploy the solution artifacts, but also covers training and enablement and handing over the application to the owners and administrators.
5. Operation – once the project is deployed, there may be administration and maintenance tasks. Think of this as the day-to-day administration and support of the solution.
6. Measurement – it is important to measure the success of the project against metrics and expected benefits. In this phase, standard and custom reports are used to run BI on the solution to see how it is doing.
7. Optimization – the project team can start to optimize the solution based on the results of the measurement. If areas for improvement are identified, they can either tweak settings and measure the effect, or return to the Analysis and Design phase is solution changes are required. (It is not necessary to start with visioning again). Usually, these subsequent iterations go faster since there are usually on minor changes or tweaks required to the solution. It is important to note that the solution will usually evolve with the business over time. It is very rare that

a BPM project is development and implemented once-off and that the solution remains in place and unchanged for years.

A Suggested K2 Project Build Methodology



Large-scale BPM projects can be quite complex for several reasons. These include the breadth of the integration with other systems, the scope of the workflow which may span across different departments and external to the organizations, and the potential impact that a business-critical process or enterprise transformational process may have.

As such, it is a good idea to have a structured and repeatable methodology to follow for these large-scale or business-critical projects. Note that it is not always required to follow this pattern especially for small, departmental-level solutions. However, if you intend to grow your BPM maturity over time and eventually implement large projects on K2, following this methodology for the initial smaller projects will allow you to build out the resources and establish the framework for the later projects.

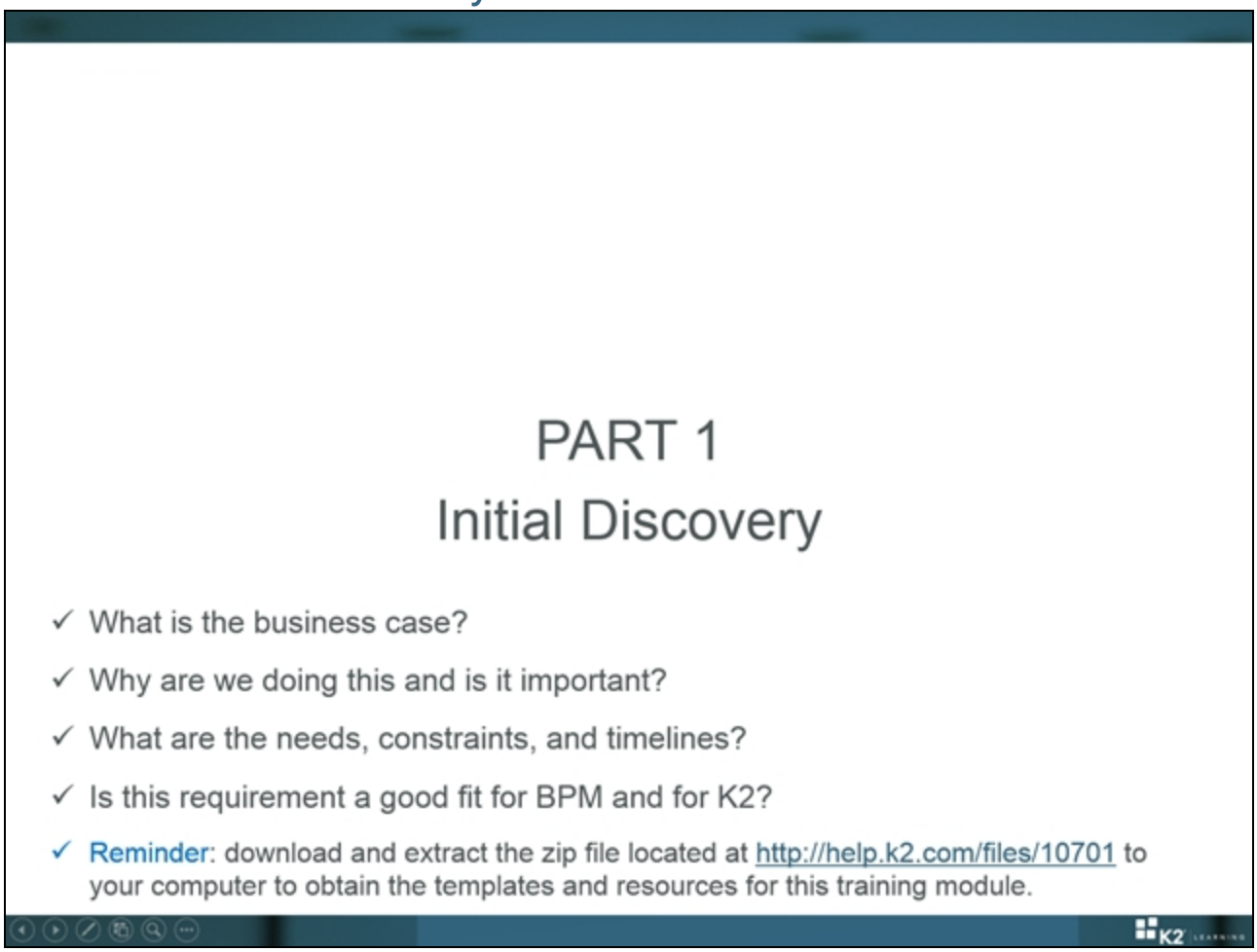
We have broken the methodology down into the phases of the BPM life cycle that are concerned with the “building” of a new application:

- Initial Discovery
- Analysis and Design
- Implementation
- Rollout

Within those are main activities and those activities have outputs. While it is not necessary to go into too much depth about the outputs right now, what we do want to point out is that this module goes from Initial Discovery to the Functional Design in the Analysis and Design phase.

This module focuses on the “Vision and Scope” and “Design” phases.

PART 1 Initial Discovery

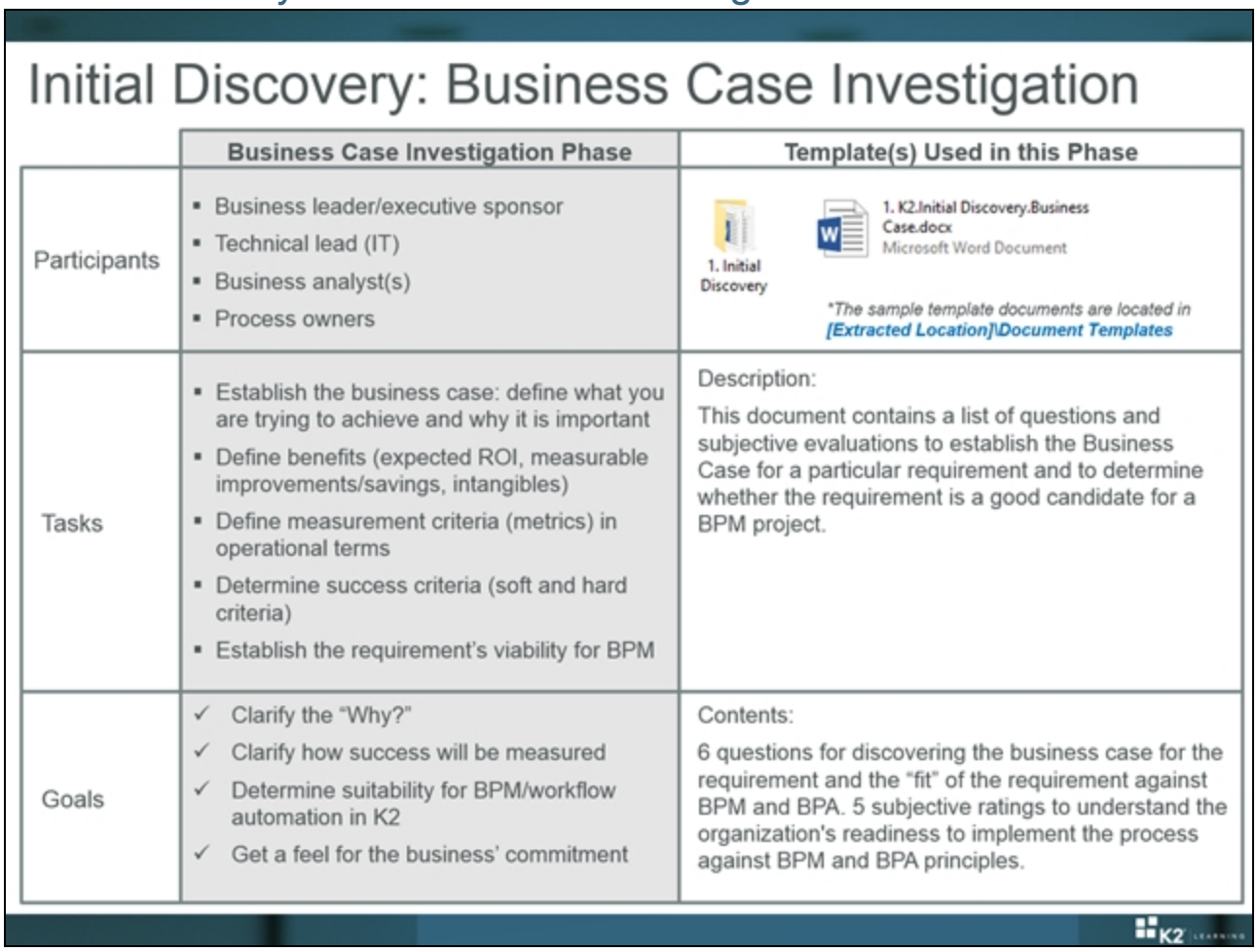


This phase is concerned with answering the following questions:

- What is the business case?
- Why are we doing this and why is it important?
- What are the needs, constraints, and timelines?
- Is this requirement a good fit for BPM and for K2?

Note
The workshop exercises in this learning module use several resources, for example template documents and workshop guides. If you have not done so already, please download the following file from K2's help site: http://help.k2.com/files/10701.
Once the zip file is downloaded, please extract the contents of the zip file to your computer. (Alternatively, you can also extract the zip file to the virtual K2 environment provisioned for you for this training course.)

Initial Discovery: Business Case Investigation



Let's look at the stages within the initial discovery first. These stages do not typically last very long: perhaps a day or two at most.

The Initial Discovery stage has two phases:

- The Business Case Investigation Phase
- The Initial Assessment Phase

Here we focus on the Business Case Investigation Phase. This phase involves four parts:

- Participants
- Tasks
- Goals
- Templates

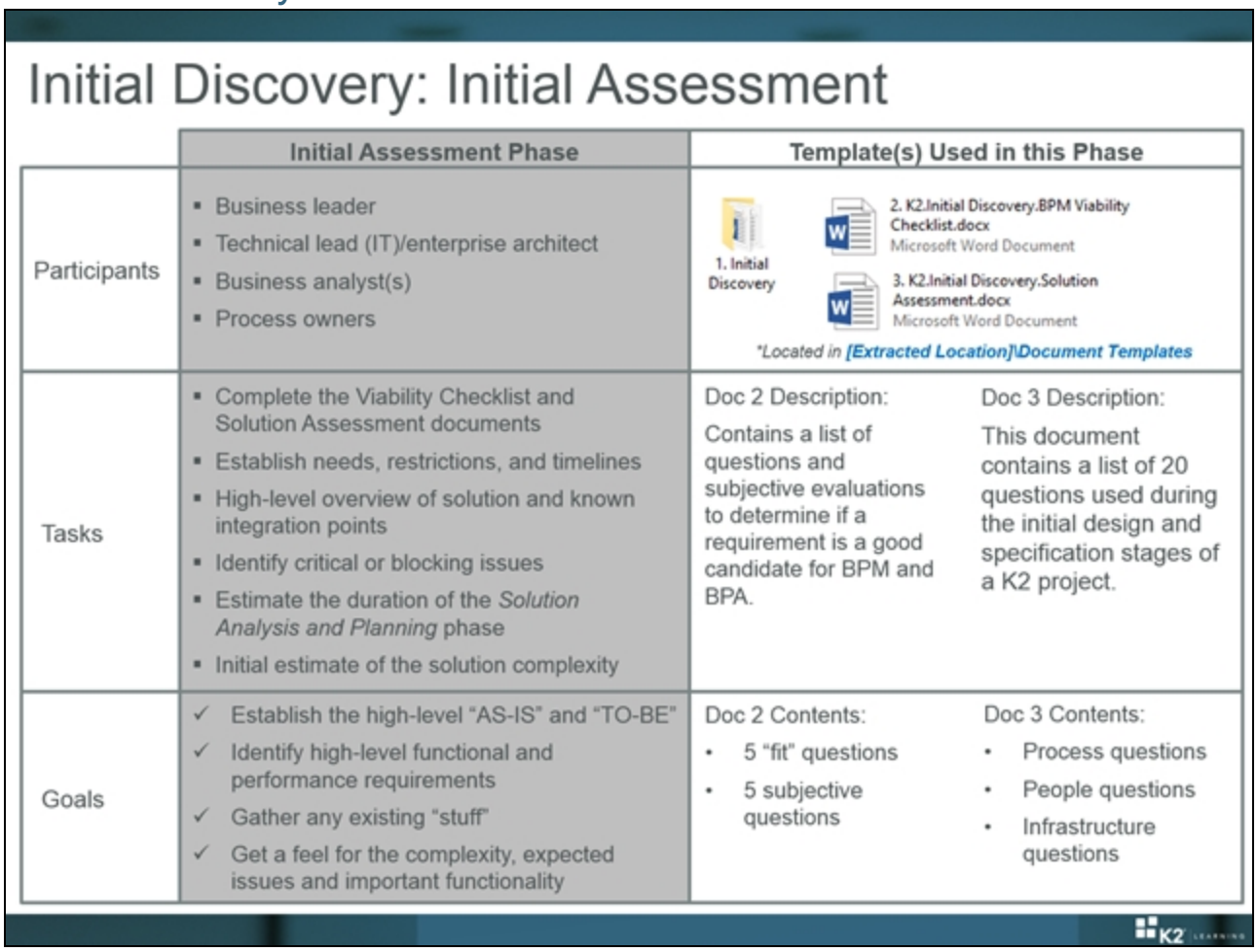
The participants use the **1.K2.Initial Discovery.Business Case.docx** template that contains a list of questions and subjective evaluations to:

- Establish the business case
- Define benefits expected ROI measurable improvements/savings other intangibles
- Define metrics in operational terms
- Determine the soft and hard success criteria
- Establish the requirement's viability for BPM

While this slide contains a lot of information about the Participants, Tasks, and Goals for the stages, fundamentally we are trying to answer the questions, "Why is this requirement important to the business?" and "Is it a good candidate for implantation on K2? Use the Participants, Tasks, and Goals described above to guide the conversations you will have with stakeholders and Subject Matter Experts during the Initial Discovery stage.

You don't have to make up any of this defining content on your own. The business case template mentioned above contains the list of questions and subjective evaluations to lead you through this phase.

Initial Discovery: Initial Assessment



The Initial Assessment Phase follows the Business Case Investigation Phase and employs a different set of:

- Participants
- Tasks
- Goals
- Templates

The participants in this phase use two different templates:

2.K2.Initial Discovery.BPM Viability Checklist.docx

3.K2.Initial Discovery.Solution Assessment.docx

The **BPM Viability Checklist** template contains a list of questions and subjective evaluations to determine if a requirement is a good candidate for BPM and BPA.

The **Solution Assessment** template contains a list of 20 questions used during the initial design and specification stages of a K2 project.

Together these two templates can help participants to:

- Establish needs, restrictions, and timelines
- High-level overview of solution and known integration points
- Identify critical or blocking issues
- Estimate the duration of the Solution Analysis and Planning phase
- Make an initial estimate of the solution complexity

This is not a complicated process. The list of participants, tasks, goals, and the templates employed in this phase gives you the framework you need to complete the initial discovery stage.

WORKSHOP SESSION 1: INITIAL DISCOVERY

Note
The workshop exercises in this learning module use several resources, for example template documents and workshop guides. If you have not done so already, please download the following file from K2's help site: http://help.k2.com/files/10701.
Once the zip file is downloaded, please extract the contents of the zip file to your computer. (Alternatively, you can also extract the zip file to the virtual K2 environment provisioned for you for this training course.)

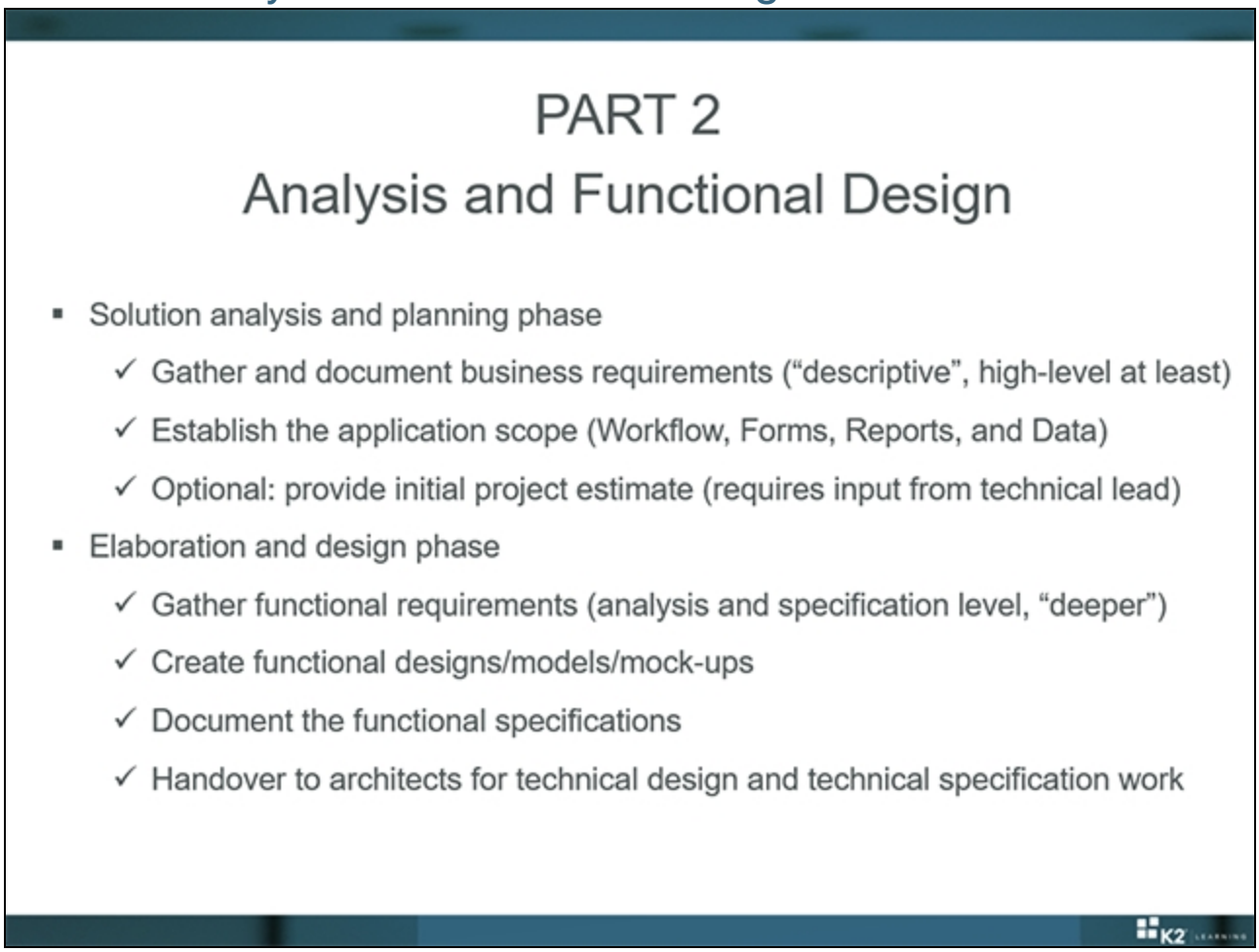
Now let's take those templates and apply them to the hypothetical expense claim approval requirements. We can use the completed document templates to help guide the conversations. After downloading and extracting the resources for this module as described in the note above, you can find the sample documents for this workshop at the following location:
[Extracted Location]\Workshop Resources\Workshop Session 1 - Initial Discovery

We will use this workshop to highlight some indicators from your own experience that suggest a good or bad fit for a K2 project. At this point, we're not going to get bogged down in technical implementation details though. Let's keep the conversation focused on determining the business case and seeing if it is a good fit for K2 and BPM.

Very often implementation decisions are made even before the requirements are known. This is the wrong way around. Typically, implementation decisions are only made once the functional requirements have been documented.

For the purposes of this workshop, we will take the hypothetical Expense Claim Approval and Processing application that the instructor demonstrated in a previous module, and then "break it down" to learn how the application was analyzed and specced. Of course, your organization may have other requirements of a similar application (or a completely different application for that matter). What is most important here is that you learn HOW to approach the Business Analysis aspect of K2 application, not the actual application that we are using to demonstrate the concepts. Feel free to apply the concepts discussed in these workshops to your own requirements as an exercise in performing Business Analysis for your own role.

PART 2 Analysis and Functional Design



Now that the initial discovery is done and we have decided this is a good candidate requirement for BPM or K2, let’s dive into the next part: Solution analysis, planning, elaboration, and design.

Here is basically what the stages boil down to:

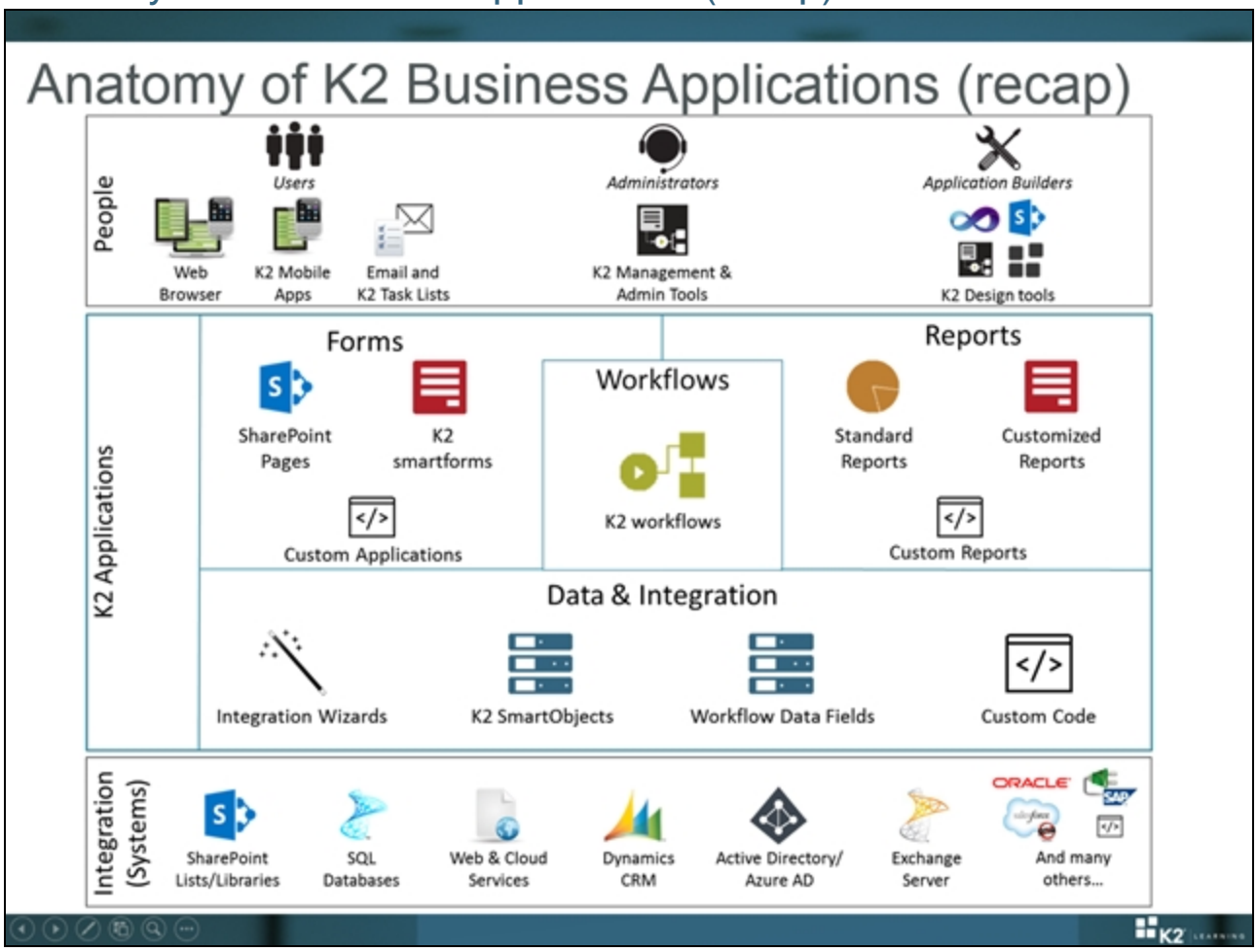
Solution Analysis and Planning Phase

- Gather and document business requirements (“Descriptive”, high-level at least)
- Establish the application scope (Workflow, Forms, Reports, and Data)
- Optional: provide initial project estimate (requires input from technical lead)

Elaboration and Design Phase

- Gather functional requirements (Analysis and Specification level, “Deeper”)
- Create functional designs/models/mock-ups
- Document the functional specifications
- Handover to architects for technical design and technical specification work

Anatomy of K2 Business Applications (recap)



The diagram above illustrates the main components of the K2 platform. We have separated these components into **People** (users who work with K2 or use K2 applications), **K2 Applications** elements (the things that make up K2 applications) and **Integration** (the systems that K2 integrates with).

K2 applications consist of four main elements: **Data**, **Forms**, **Workflows** and **Reports**. The application elements fully integrate with each other to provide a robust framework for building applications. However, they also work independently from each other. For instance, you might have a SharePoint list that uses SmartObjects and SmartForms, but not Reports or Workflows. K2 also provides tools for designing and customizing the application elements as well as tools and interfaces to administer the K2 environment, such as managing permissions and managing the active workflows in the environment.

People, Roles, and Administering K2 Applications

K2 defines a number of logical roles that make up the human element of your Application. Users are the consumers of your Application's functionality, and they use the Forms that you build to interact with your Application. Usually, users would use your applications with web browsers or mobile applications. Administrators manage workflow instances and permissions, and use K2's Management Pages or other administration tools to perform these functions. Application Builders are the designers and builders of the Forms, Data, Workflows, and Reports using the web-based tools such as the K2 Designer.

Forms

Forms are the User Interfaces that let users interact with your application. K2 supports a range of technologies for these Forms: K2 smartforms, InfoPath Forms, SharePoint interfaces, or custom applications written in .NET or other programming languages. The exact technology used for an application depends on the requirements and the organization's infrastructure, but the main point is that K2 gives you the freedom to choose a technology for creating forms.

Workflows

Workflows are a logical sequence of steps and events performed by users and/or systems. Workflows provide the "business process" functionality in your application. K2 features a mature and powerful workflow engine which has proven itself over many years in many different organizations.

Consider something like a leave request application. In this case, there would be a workflow component which runs the leave request-approval-processing steps of the application. The workflow assigns tasks to users (the employee's manager, for example, when the leave request must be approved) and performs system tasks as well (for example, sending e-mail notifications to the employee to notify them that their leave request was approved).

Reports

K2 provides standard reports that expose the metrics of your application workflows, such as the time taken to complete leave requests and audit trails. You can also use SmartForms and SmartObjects to create customized dashboard-style reports, leveraging the available reporting controls in the K2 Designer tool. Or you can build custom reports using third-party reporting tools that are able to consume the available K2 APIs and services.

Data and Integration

The Data component allows you to integrate your application with providers of data and consumers of data. K2 primarily uses a technology called SmartObjects to interact with back-end systems. You can think of SmartObjects as a "connector" or a "middle layer" that provides the necessary integration to expose the data that resides in some system, to the K2 application elements that need to use that data. The important point is this: SmartObjects are the consistent interface that K2 uses to interact with some provider of Data, regardless of what technology that provider might be. The artifacts in that provider (for example the "Employee" or "Customer" or "Account" entities) are represented as logical business entities that you can easily use within your applications, without having to know where that data resides.

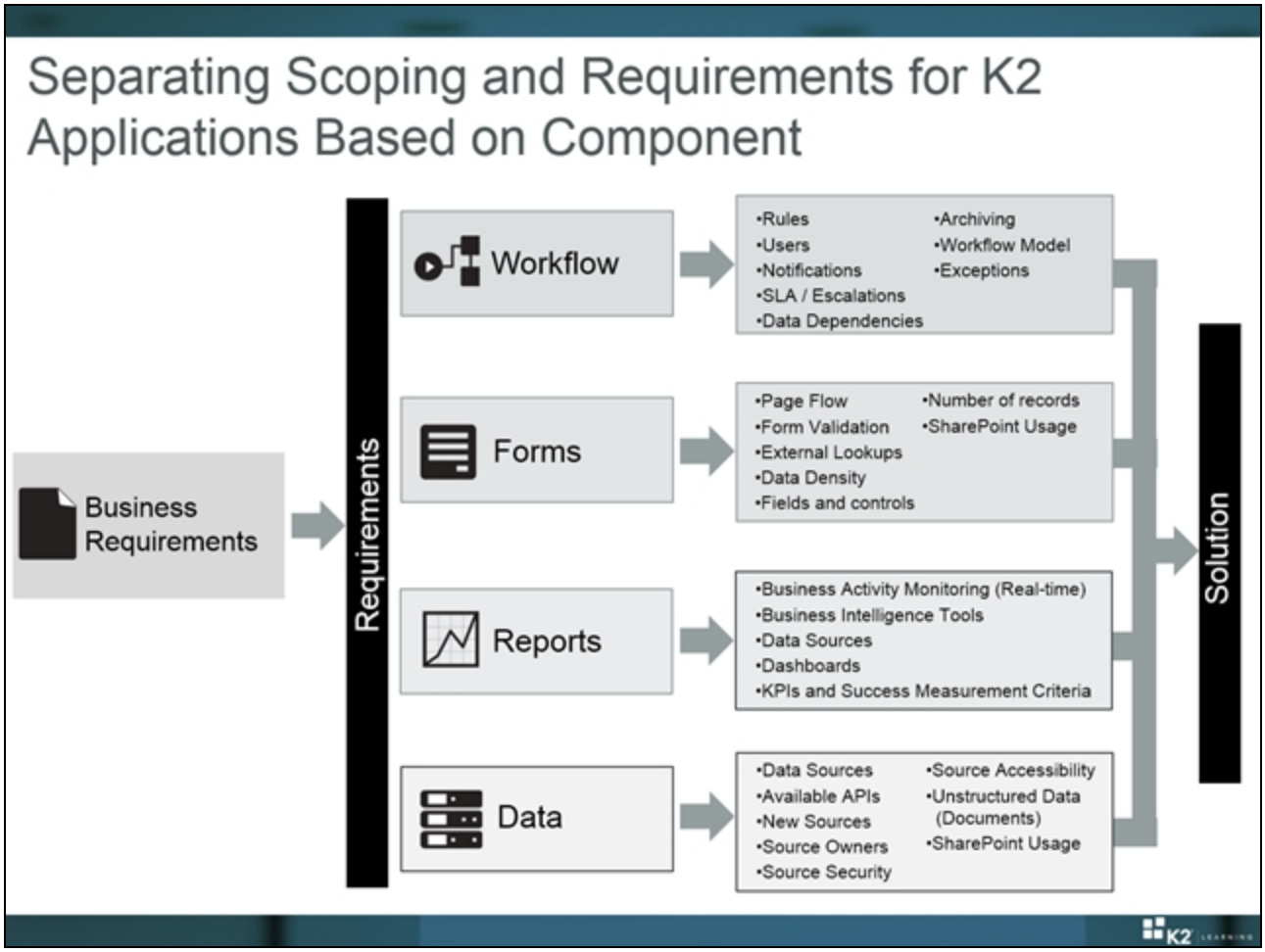
Data could also refer to workflow-level data fields that are defined in workflows, but this is normally limited to storing data that is only applicable to the workflow itself.

K2 provides a set of standard workflow wizards that make it easy to integrate with common enterprise systems like Active Directory, Exchange, SharePoint, and many others.

Summary

- K2 applications consist of Forms, Reports, Workflows, and Data (Data includes integration with LOB systems).
- The most common options for each components, are SmartForms for the Forms part, SmartObjects for the Data part, and standard reports for the Report part.
- SmartObjects are used as a Data Access layer to integrate with other systems.
- Users always interact with Forms or Reports. They would use browsers, K2 applications, email or the K2 task list. Users could also be using custom applications if the organization built their own applications.
- Administrators use the K2 Management Pages to administer K2 environments.
- Application builders use K2 web-based design tools, K2 thick client tools like K2 Studio or tools like K2 for Visual Studio to build applications.

Separating Scoping and Requirements for K2 Applications Based on Component



By breaking K2 applications down into Workflow, Forms, Reports, and Data it becomes a little easier to separate out the requirements (and eventually, the development as well). Think of it as a divide-and-conquer approach: the requirements that result in full-blown K2 projects are often very complex and deeply integrated, so by breaking the requirements discovery up into smaller chunks it becomes easier to start gathering requirements.

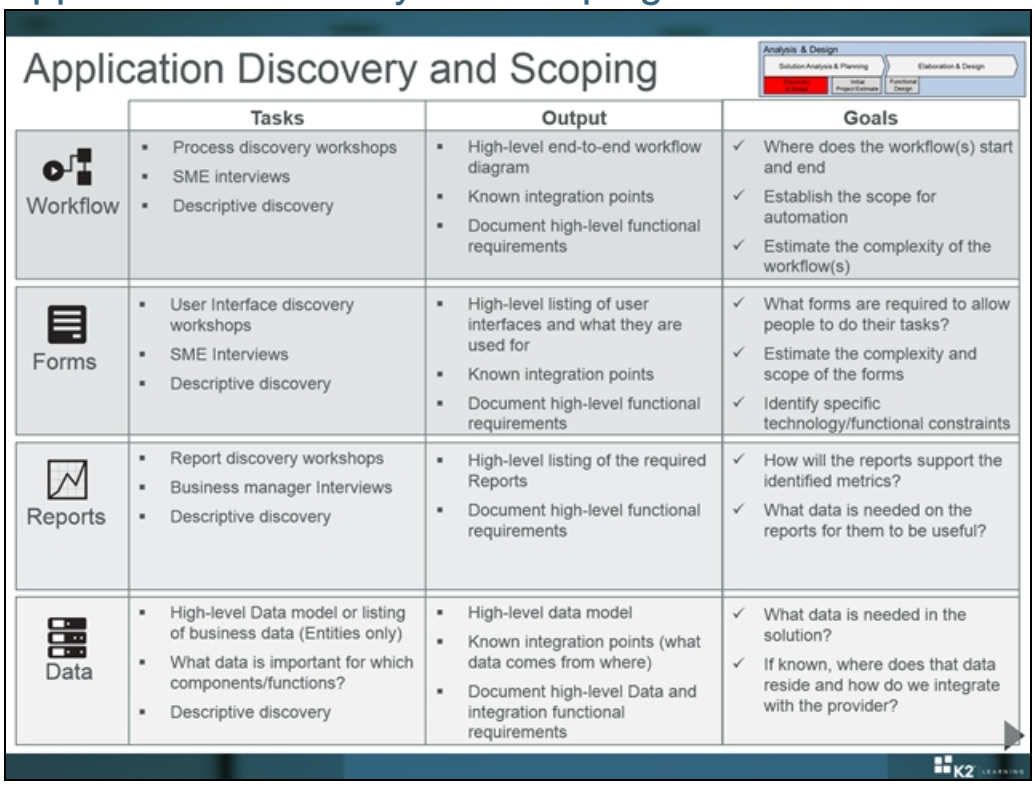
It is also easier if you approach it in the sequence suggested here:

- start with Workflow
- then go to Forms
- do the Reports
- finally do the Data

There are several reasons why we recommend breaking up the scoping and requirements gathering :

- It allows a "separation of concerns" approach where different components perform different tasks in the application
- It allows multiple teams to work on different components at the same time, which can help to speed up discovery and requirements gathering
- since BPM projects can be very complex, it helps by reducing the scope of the application to a specific component without getting diverted into other components
- typically, K2 applications are implemented with separate Workflow, Forms, Reports and Data components, so this approach helps to keep everything aligned from requirements gathering through to implementation

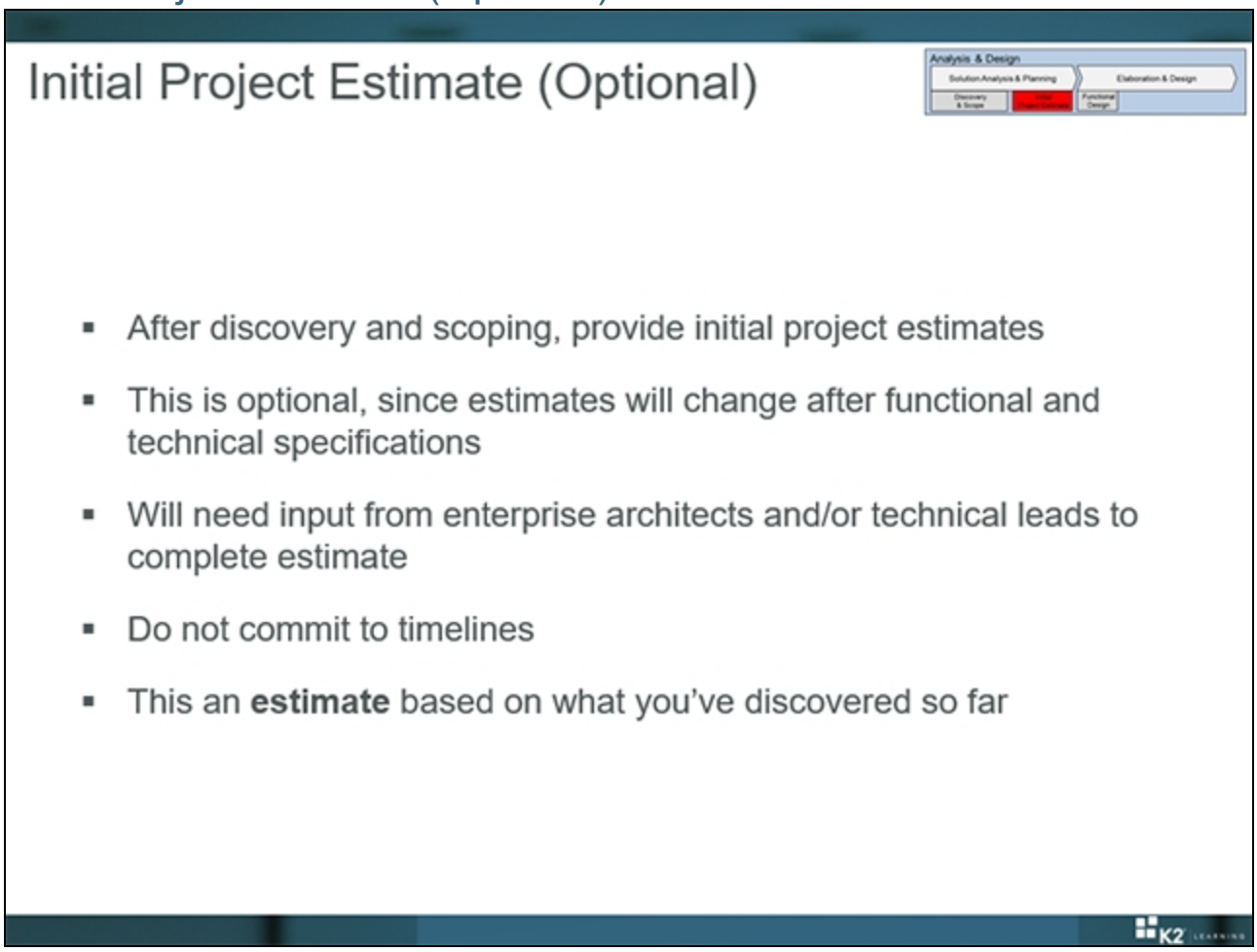
Application Discovery and Scoping



At the end of the initial discovery and scoping stage, you should have a better awareness of the high-level requirements for the application. This will help if you need to provide an initial estimate for the application.

Later on we will look at some document templates that will help you limit the discussions to the high-level requirements necessary to complete this phase. For now, just keep in mind how to separate out the components and what the goals are during the application discovery phase. You may also want to establish what the application should and should not do. Think of this like drawing a line in the sand: this is what we will implement. This can be difficult, but it is important to get clear understanding and agreement of what exactly the application will and will not do.

Initial Project Estimate (Optional)

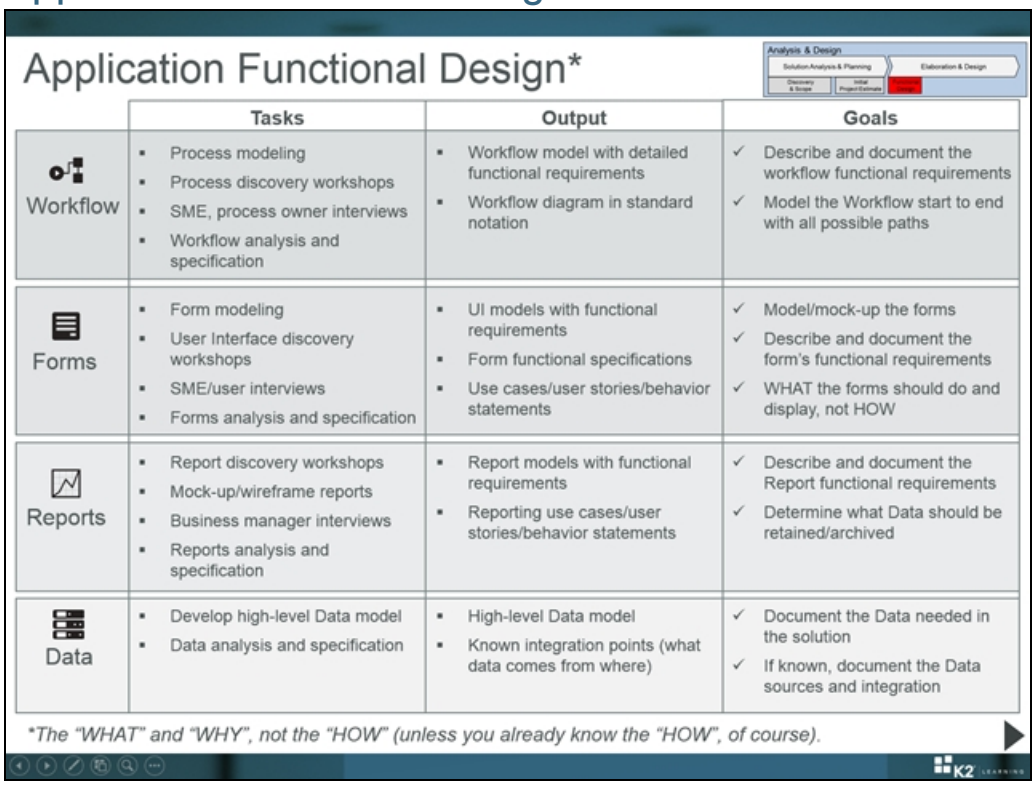


Now that you have established the high-level requirements and initial scope, you can provide an initial estimate.

Note that this is optional, since the estimates will typically change after the technical design is completed. However, we all know that people want budgets and timelines. This is an estimate only, intended to give a ballpark figure of the duration and complexity of the project. Do not commit to timelines since they will have to change anyway once the technical design is done.

Note that you should solicit input form the enterprise architects and/or technical leads during this estimate, because the template we provide helps you start breaking the application down into separate components.

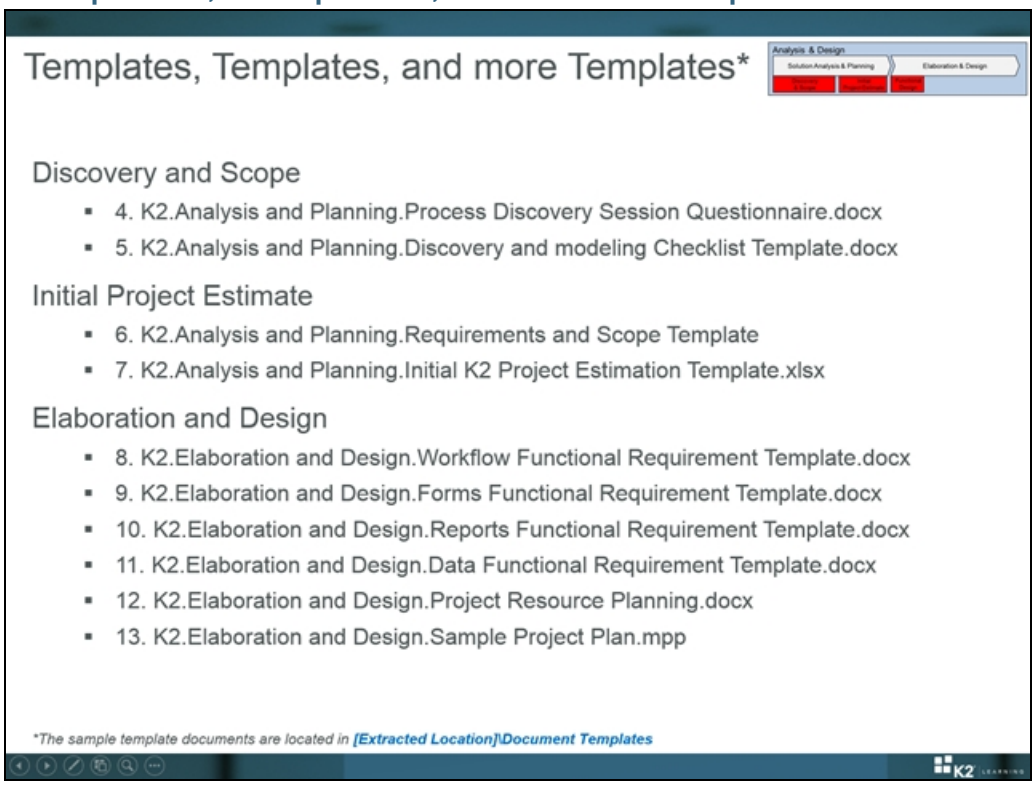
Application Functional Design



Now we move on to the Functional Design stage. This is where things get more interesting (or less, if you hate checklists of questions).

Completing the process, form, report, and data modeling activities, shown here under Tasks, will provide the framework and details you will need to move ahead with development. Essentially, at the end of functional design you should have a set of documents that describe the functional requirements that provide models of the Workflow, Forms, Reports, and Data for the application. These are then fed into the technical team for the architecture, design, and technical specifications of the application.

Templates, Templates, and More Templates



If you like templates, we have a lot of templates that you can use during the Analysis and Design phase of your project to gather and document requirements.

You do not have to use these templates, and you are free to modify these templates to suit your own requirements. However, using these templates will help to ensure that you are asking the right questions at the right time and not getting too deep into the technical intricacies of implementation.

Here is a brief summary of each template.

4.K2.Analysis and Planning.Process Discovery Session Questionnaire.docx

This is a sample questionnaire to help participants in a process discovery session prepare for the session. It is not necessary to answer every question in detail during the session, but at least have a high-level explanation to the listed questions.

5.K2.Analysis and Planning.Discovery and modeling Checklist Template.docx

This document contains a list of suggested questions intended to help the flow of the various discovery sessions that are typically held during the Analysis stage of a K2 project.

In general, the questions flow from Descriptive to Analytical to Specification:

- Descriptive: general information gathering, the who, why, and what
- Analytical: more specific questions aimed at answering the how
- Specification: even more specific questions aimed at discovering exactly how

6.K2 Analysis and Planning.Requirements and Scope Template

This is a document to capture a proposed K2 application's functional requirements and scope. It is typically completed during the Solution Analysis and Planning phase. Use this Requirements Specification template to collect, organize, and record the requirements for your project, including priority and approval. Tailor the specification to suit your project, organizing the applicable sections in a way that works best, and use the checklist to record the decisions about what is applicable and what isn't.

For ease of implementing the requirements and tracking with subsequent design and specification documents, we recommend separating requirements into Workflow, Forms, Reports, and Data.

7. K2.Analysis and Planning.Initial K2 Project Estimation Template.xlsx

This template is an Excel spreadsheet with three tabs:

- SA&P Estimator
- Task Factors
- Time Factors

The **SA&PP Estimator** is structured in the following categories:

- Infrastructure Services
- Build
- Integration and Extension
- Workflow
- User Interface
- Reporting
- SharePoint
- Misc

The **Estimated Hrs** cells in each category have been set up with sum formulas that calculate totals at the bottom of this sheet.

The **Task Factors** sheet allows you to fill out task detail for each of the categories listed in the SA&PP Estimator.

The **Time Factors** sheet contains base numbers used in calculations. You can change these base figures to suit your own organization.

8. K2.Elaboration and Design.Workflow Functional Requirement Template.docx

This document is a sample template that you can use to gather requirements and specifications for the workflow(s) component of K2 applications. It is normally completed by Business Analysts after the workflow discovery has been completed. You can use it as a basis for extending the technical specification for the workflow in larger projects, or as the technical specification developing the workflow for simpler and smaller K2 workflow solutions, if the details of the workflow steps have been captured using the templates in the Appendix section.

9. K2.Elaboration and Design.Forms Functional Requirement Template.docx

This document is a sample template that you can use to gather requirements and specifications for user interfaces (also known as UIs or Forms) in K2 Projects. This document is usually accompanied by other documents that cover the Workflow, Reports, and Data Functional requirements.

10. K2.Elaboration and Design.Reports Functional Requirement Template.docx

This document is a sample template that can be used to gather requirements and specifications for Reports and business intelligence components in K2 Projects. It is usually accompanied by other documents that cover the Workflow, Forms, and Data functional requirements.

11. K2.Elaboration and Design.Data Functional Requirement Template.docx

This document is a sample template that you can use to gather requirements and specifications for Data and Integration in K2 Projects. It is usually accompanied by other documents that cover the Workflow, Reports, and Forms functional requirements.

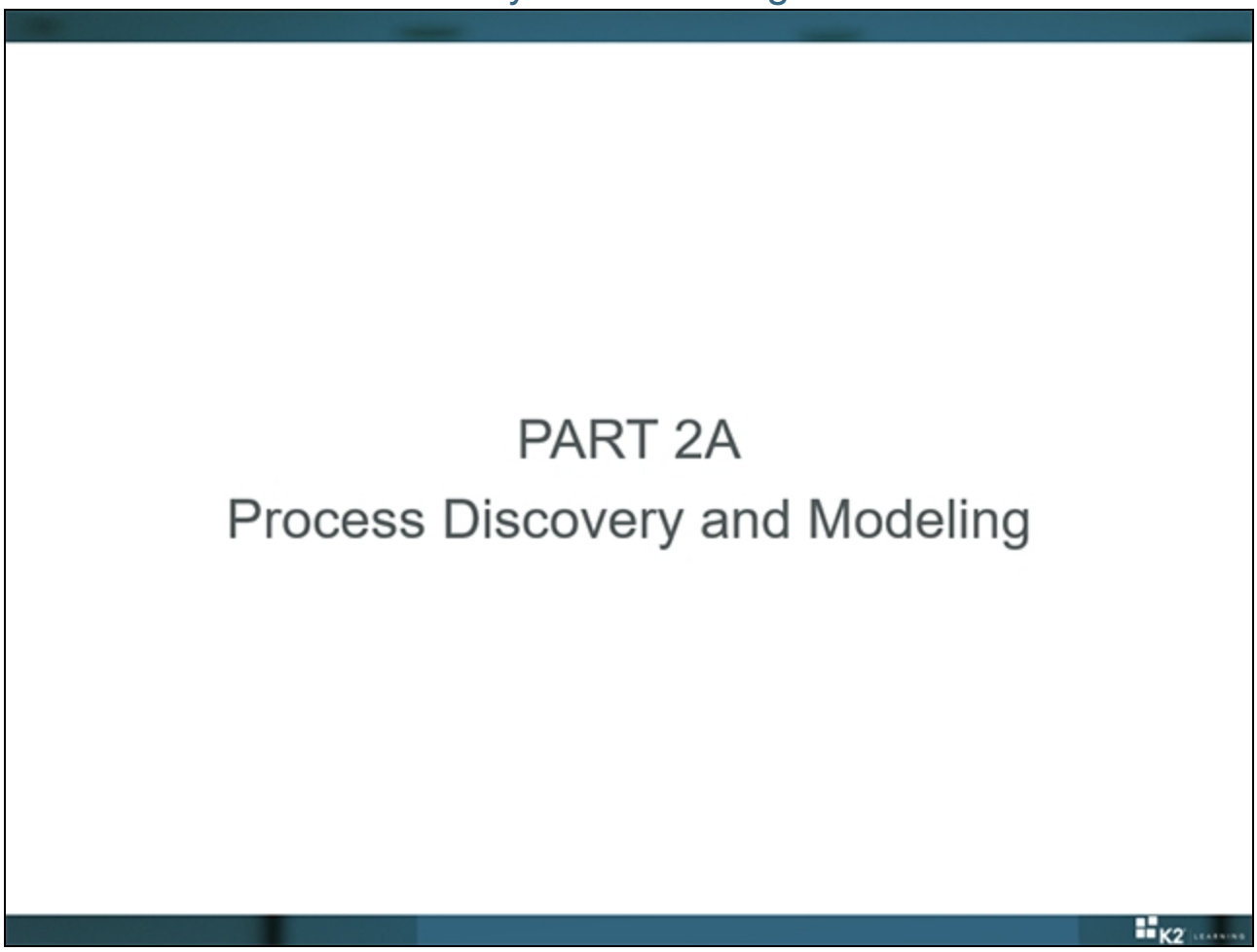
12. K2.Elaboration and Design.Project Resource Planning.docx

This document can help you identify the various roles commonly found within a K2 blackpearl environment and in project-level solutions. A roles table outlines and lists responsibilities and skills for the most common roles. This template is intended to provide general guidance based upon scenarios found at many K2 customers, and even internal roles and skills at K2. Use it to evaluate the guidelines within the context of your operating environment and factor in your own policies, infrastructure, personnel, and timelines. Once this evaluation is complete, you should be able to articulate a defined set of roles that can be mapped to your organizational structure.

13. K2.Elaboration and Design.Sample Project Plan.mpp

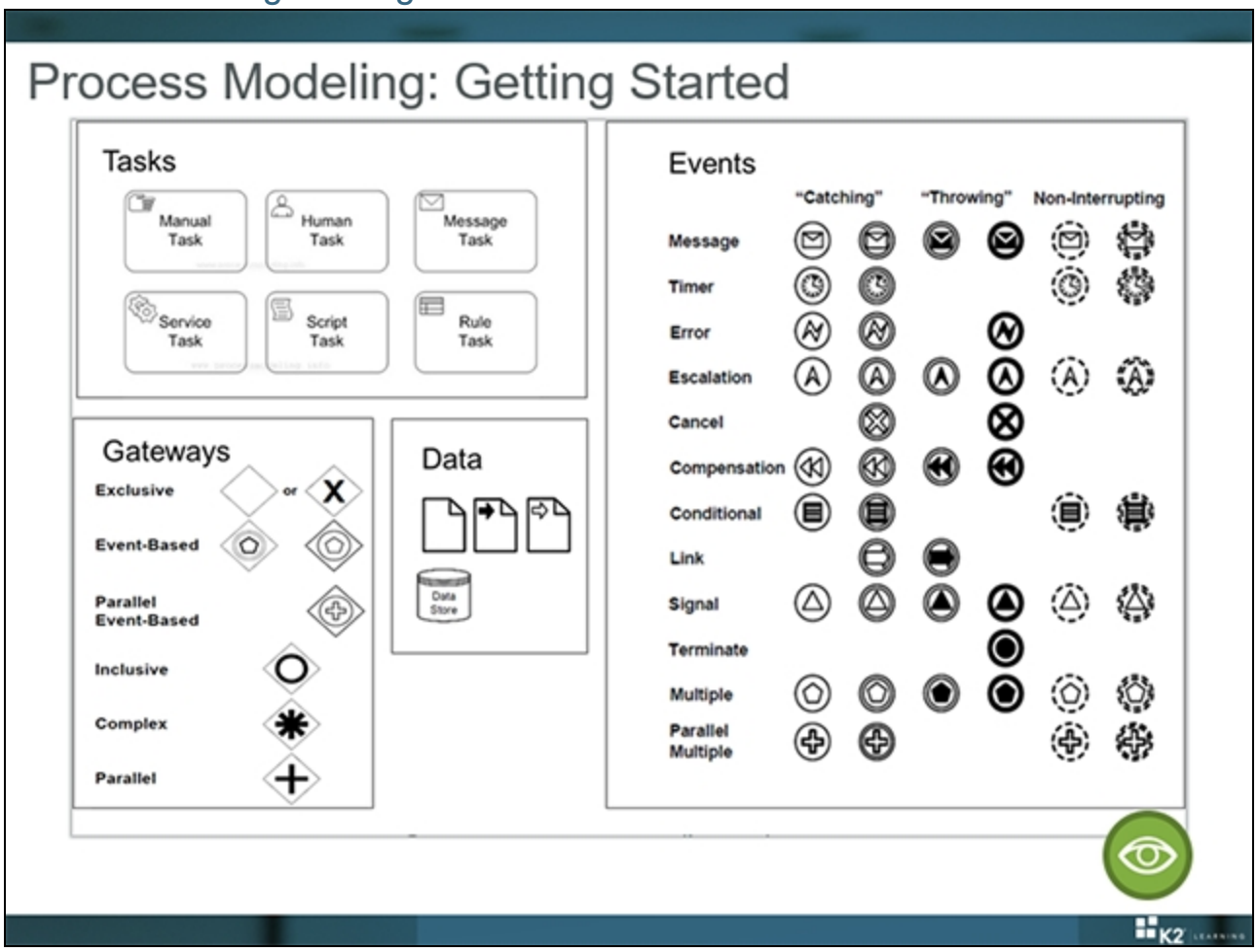
You can adapt this Microsoft Project sample project plan to your own needs and circumstances.

PART 2A Process Discovery and Modeling



Now let's look at how you model processes. This is important, because good modeling can really help to ensure that the end result does what is expected. Complex processes can devolve into a huge mess, so try to keep things on the path we recommend, especially for larger more complex processes.

Process Modeling: Getting Started



When dealing with large projects and requirements, it can be difficult to know where to start due to the sheer number of moving parts. Just break it down into steps.

Start with the process and then, for that process, start with the basics:

- Name of process
- Process owner
- Process scope
  - How does it start? What are the triggering event(s)?
  - When does it end? What is the output or result?

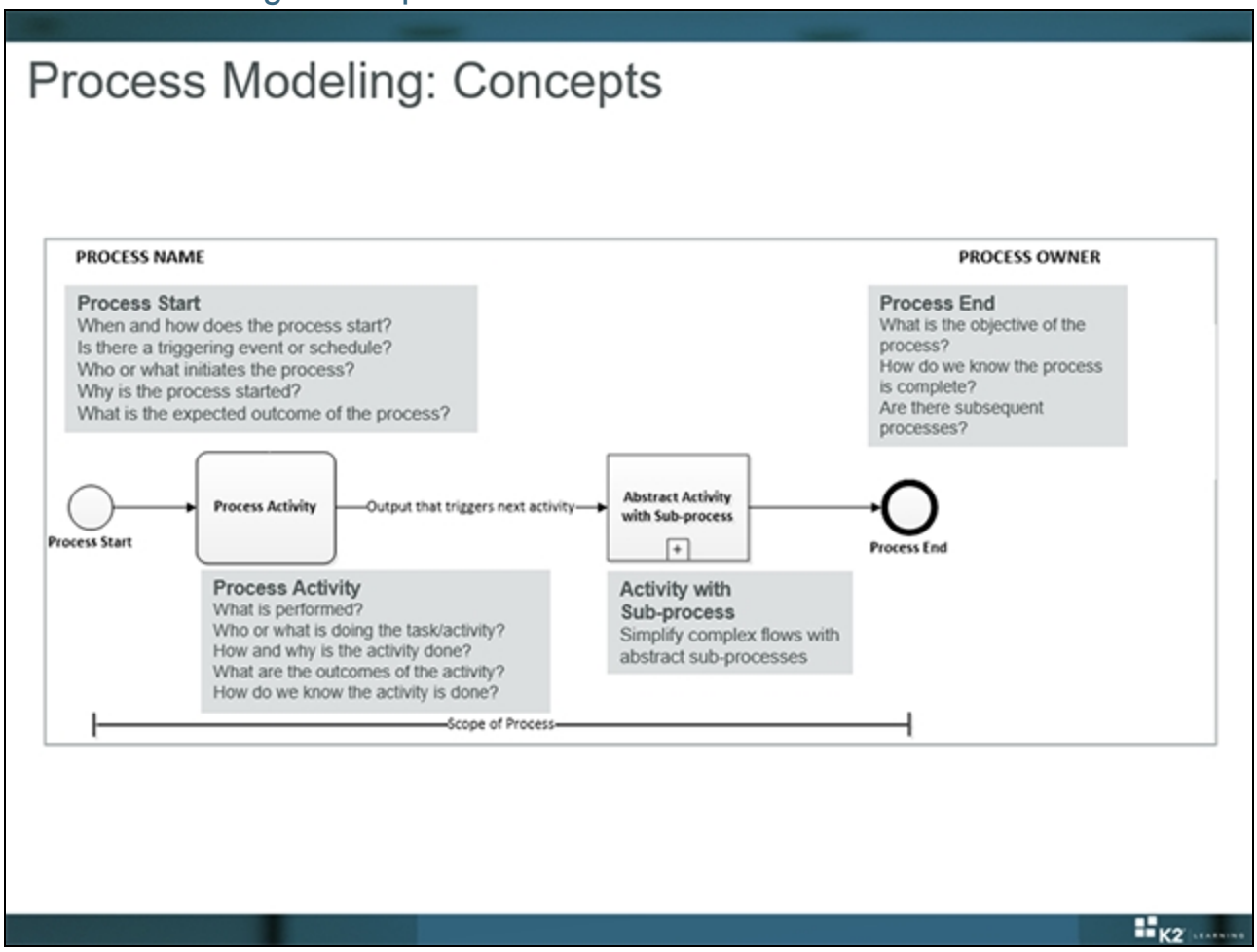
Next look at the steps in the workflow:

- Activities and flow
  - Establish the high-level breakdown of the process steps
- Drill-down to detailed definitions for each activity/step
  - Who? What? How? What Data? Outcomes?

Some hints to help with getting started modeling processes:

- Start with the “happy” path
- Iterate for exceptions and escalations
  - Add more details to the process model to make it more explicit
- Use consistent notation
  - Consider using the BPMN standard (we have provided a poster of BPMN notation)
  - You do not have to use BPMN for K2 workflows (unless you want to export-import BPMN models) but it helps to use a commonly-understood modeling language.

Process Modeling: Concepts



This simple model shows a business process in its most basic, stripped down essence. It has fundamental process components:

- Process Start
- Process Activity
- Abstract Activity with Sub-process
- Process End

Each node or component above also has sample questions that can help you define it. For example, let's look at the **Process Start** node questions.

- When and how does the process start?
- Is there a triggering event or schedule?
- Who or what initiates the process?
- Why is the process started?
- What is the expected outcome of the process?

When and how does the process start?

Processes have a stimulus that initiates them and a timing factor.

Is there a triggering event or schedule?

Identify the stimulus that initiates the process. Is it a periodic event on a timetable?

Who or what initiates the process?

This question helps identify whether the stimulus is human or technological in origin.

Why is the process started?

This question speaks to the business reason for the process.

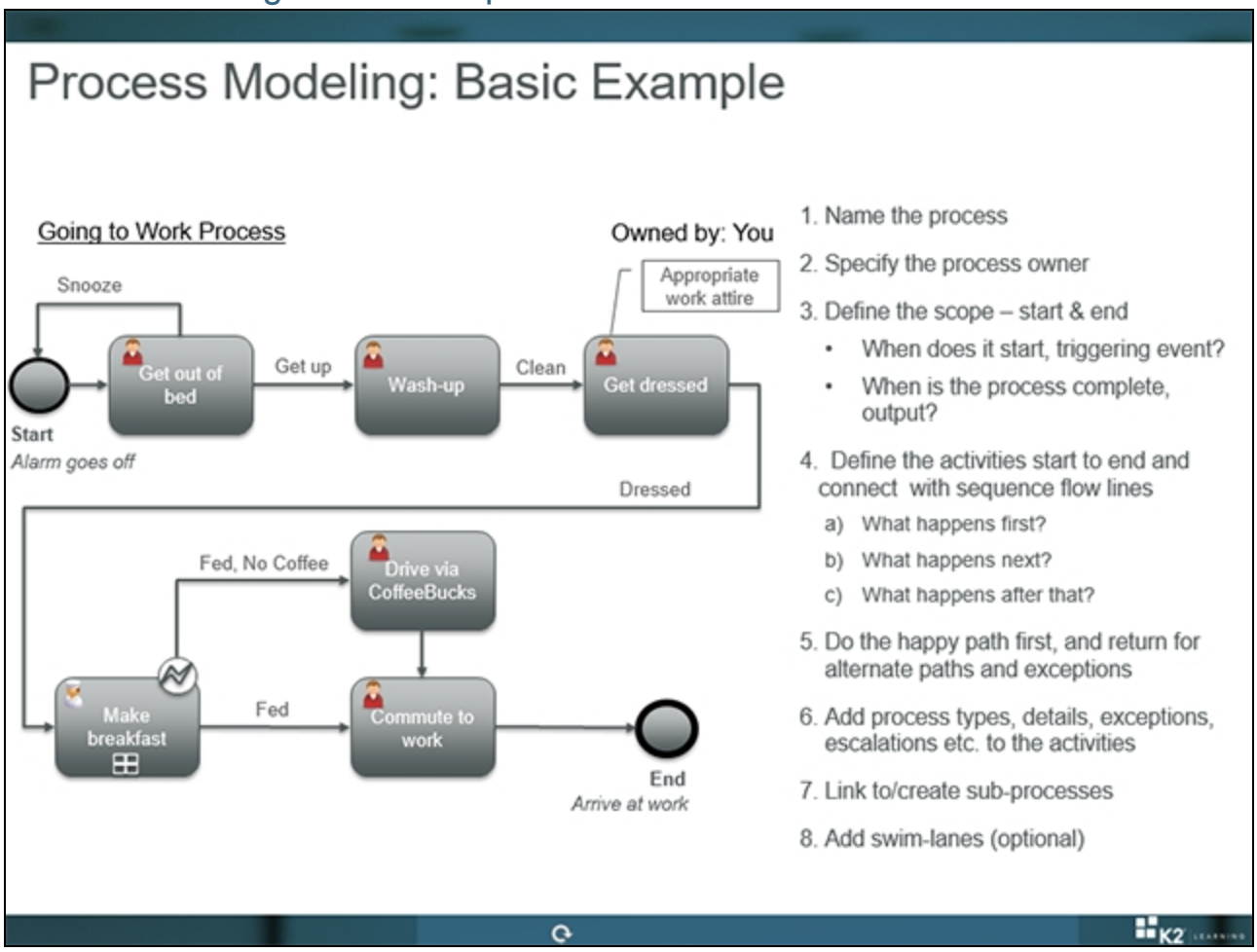
What is the expected outcome of the process?

You should be able to describe or define what is expected as a result of this process.

While there appears to be some degree of overlap between these questions, each one can help you more clearly understand the purpose of the node.

Outputs between nodes join activities together, and the scope of the process runs from the start to the end. Basic stuff, perhaps, but sometimes you have to start from the basics.

Process Modeling: Basic Example

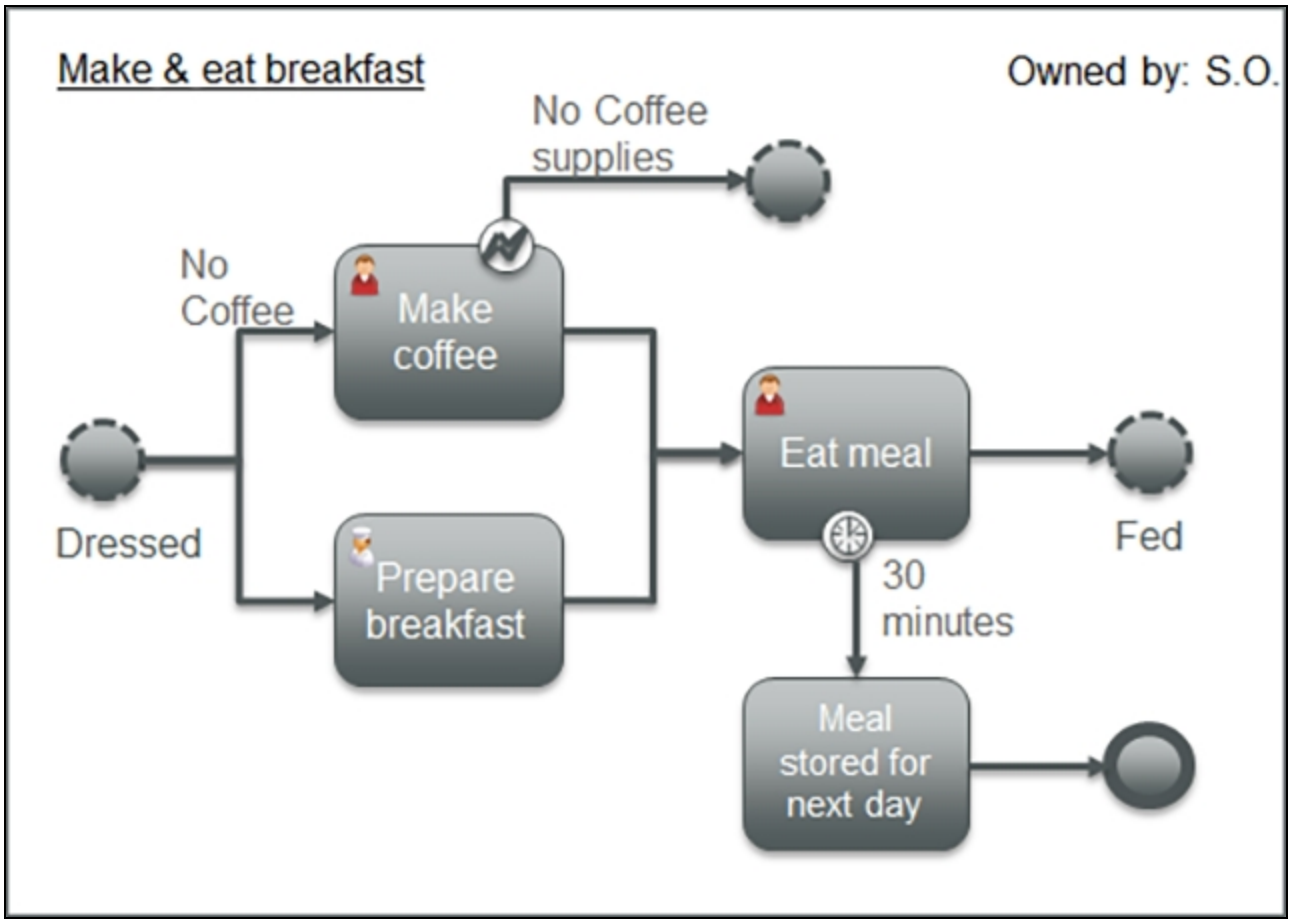


To put process modeling into practice, we will do a short exercise on a common process: getting up and going to work.Think what this process would look like for yourself in your own typical workday, and draw out your own daily “workflow” if you like.

Essentially, we want to learn how to approach process modeling:

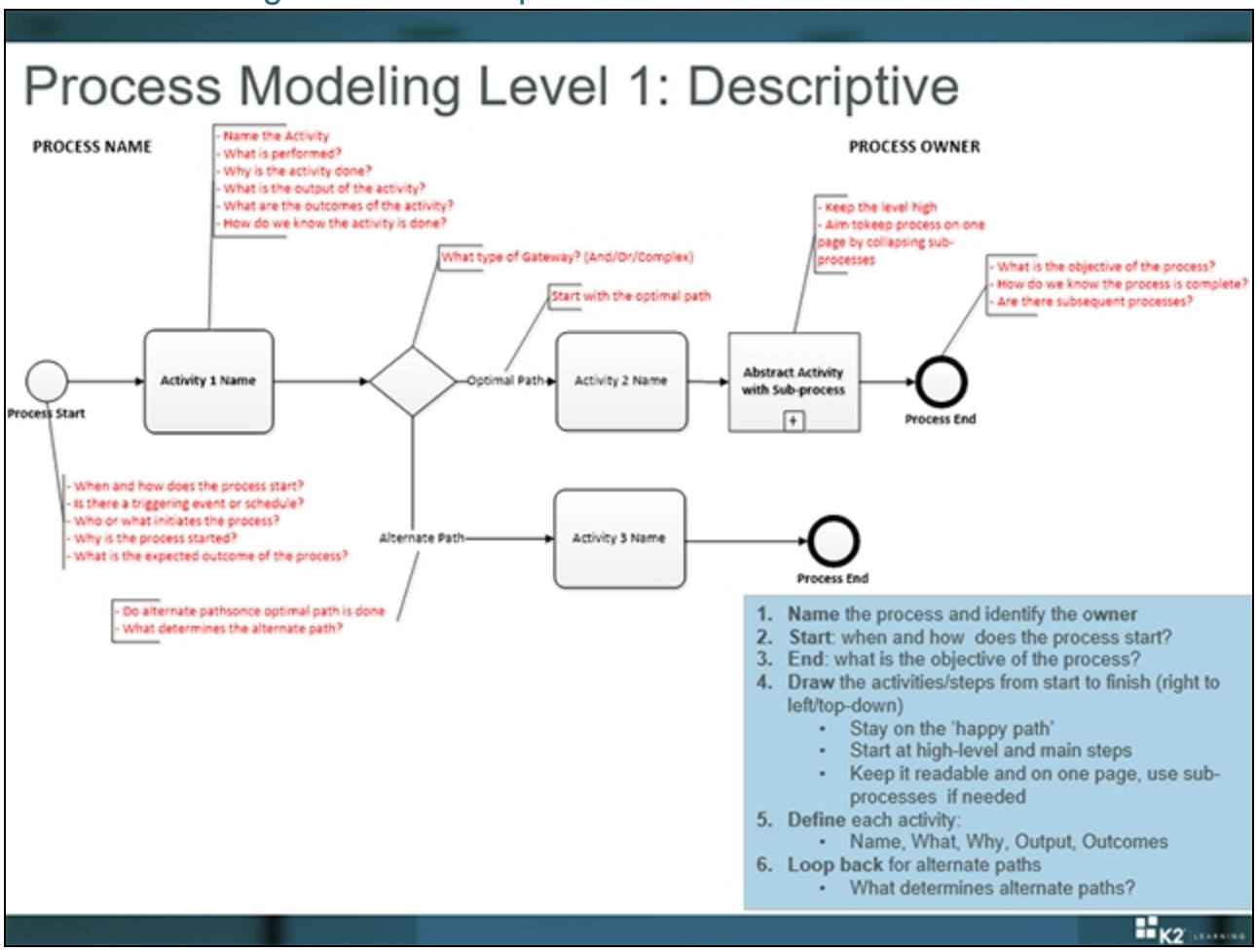
1. Name the process. (Going to Work Process)
2. Specify the process owner. (You)
3. Define the scope – start and end.
    - When does it start, triggering event? (The alarm goes off)
    - When is the process complete, output? (Arrive at work)
4. Define the activities start to end and connect them with sequence flow lines.
    a. What happens first? (Get out of bed)
    b. What happens next? (Wash up)
    c. What happens after that?
    (Get dressed, make breakfast, either drive to coffee shop or commute to work)
5. Do the **happy path** first, then return for alternate paths and exceptions.
    - The alarm goes off at the right time.
    - There is clean clothing for dressing.
    - There is something in the fridge to prepare for breakfast.
    - You have coffee at home. (etc.)
6. Add process types, details, exceptions, escalations etc., to the activities.
7. Link to/create sub-processes.
8. Add swim-lanes (optional).

Make and East Breakfast Process Diagram



You might think of the Make and Eat Breakfast workflow as a "sub-routine" of the Going to Work Process. In it you can see more detail that wasn't specified in Going to Work. When you define a workflow, the "granularity" of process detail may be an important factor to consider.

Process Modeling Level 1: Descriptive



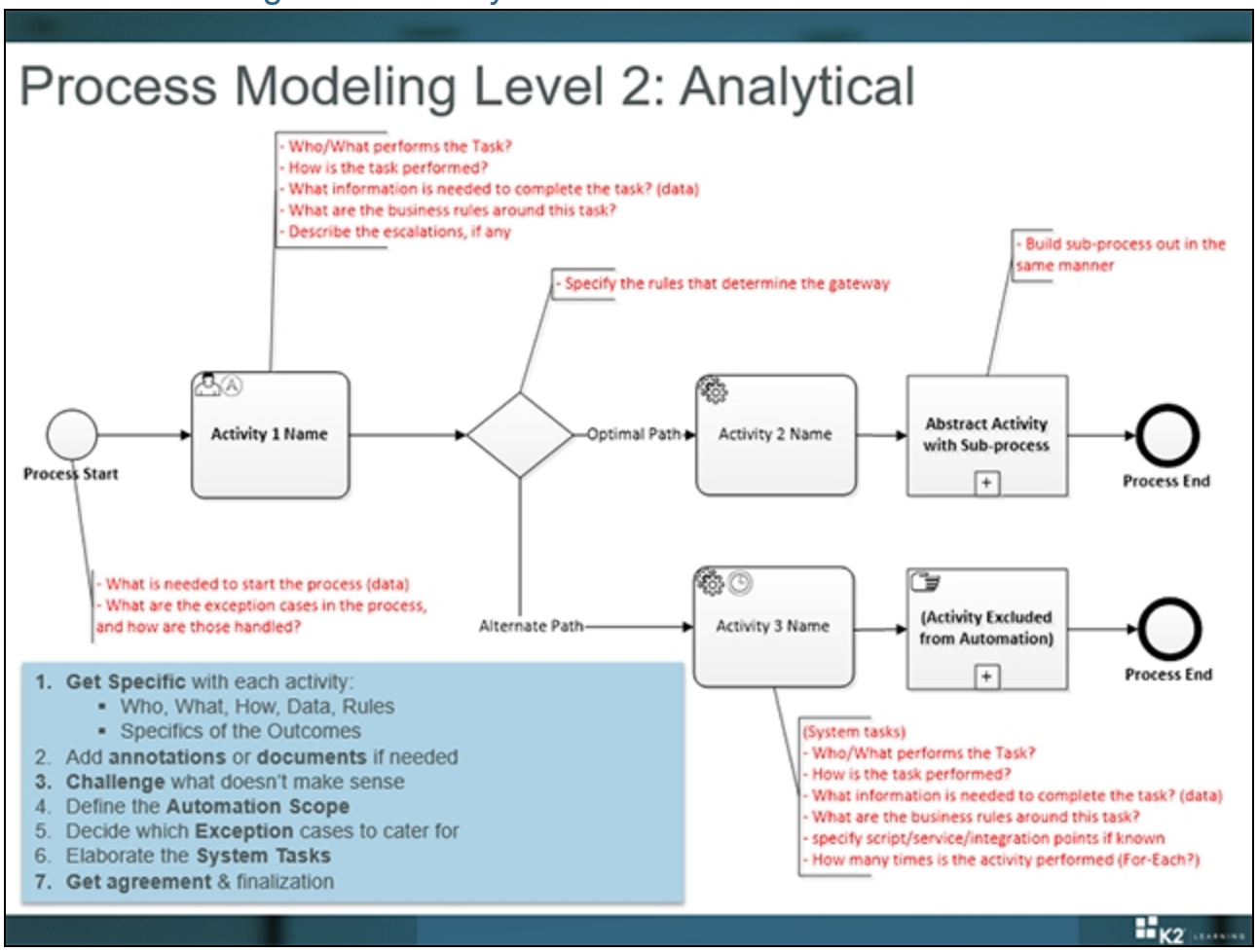
When modeling workflows you can use a technique known as Descriptive-Analytical-Specification to start with the basics and drill down. The following slides illustrates Analytical and Specification phases and the typical questions you would ask during these phases. Notice as you progress through the following Analytical and Specification steps that the process is based on the process shown here. Here we begin with the **Descriptive** phase of this method.

The discovery questions we have provided in the template **5. K2.Analysis and Planning.Discovery and Modelling Checklist Template.docx** follow these principles. **Section 1.1** of the template document contains the:

- Questions for the overall process
- Directions for diagramming your workflow steps and alternate paths
- Questions for the common path
- Questions for the alternate paths

These questions are shown in red in the diagram above. When you are scoping the application, it is a good idea to have completed at least the Descriptive phase of modeling before you work on the scope and system requirements. That way you have at least done the initial discovery of the process. Following the scoping, you can then return and complete the Analytical and Specification phases of the process modeling.

Process Modeling Level 2: Analytical



In the **Analytical** phase you will look deeper into the business rules. This phase corresponds to section 1.2 of the **5. K2.Analysis and Planning.Discovery and Modelling Checklist Template.docx** template titled Analytical Discovery and Modelling (Workflow).

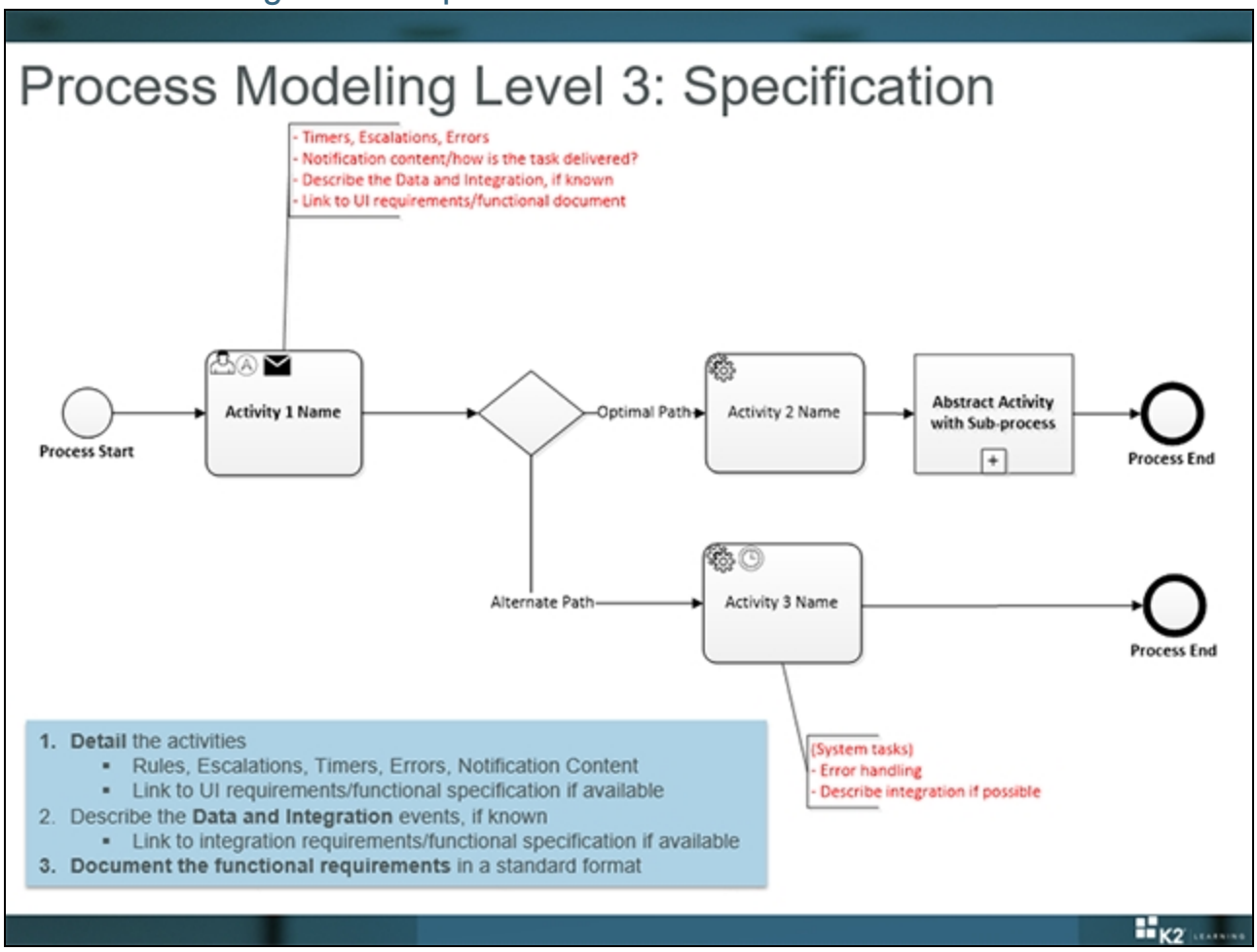
Here we want to discover the specific details of the steps in a workflow such as who does the work, how they do it, and what information they need. You may start doing the Forms discovery at the same time, but it is usually a good idea to finish the workflow model before moving on to Forms.

Consider the level of detail achieved in this phase by reviewing some of the questions from section 1.2 of the template.

- Is there some kind of reference number, unique identifying information or other way you would distinguish between instances of the workflow? Is this a combination of values (e.g. “customer name” + “order number”)
- What are the other exception cases in the process, how are those handled, and how often do they happen?
- What are the escalation/SLA consideration for the process as a whole?
- Who is allowed to Report on all instances this workflow?
- Is there a reason to limit certain people/roles to only report on the instances of the workflow they were involved in and if so, who/what are those people/roles?
- Who would administer this workflow, for example performing manual overrides, manual task redirection or pausing/resuming, if that was necessary?

This phase is also where you should get agreement and finalize the business aspects of the process.

Process Modeling Level 3: Specification



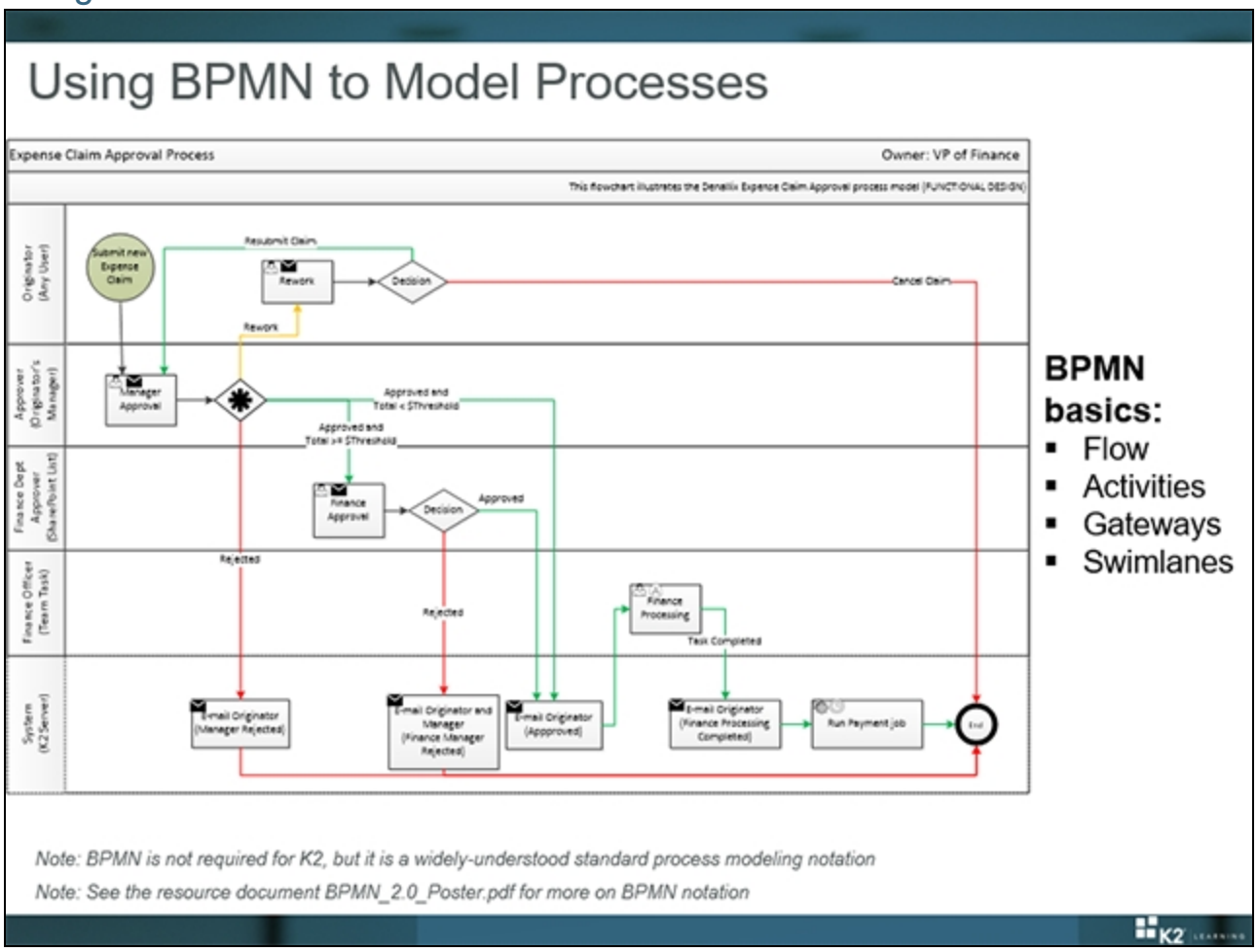
In the **Specification** phase you are getting into the practical details of process modeling. This phase corresponds to section 1.3 of **5. K2.Analysis and Planning.Discovery and Modelling Checklist Template.docx**. This is where you start to specify exactly how things are supposed to work. While this template supplies standard questions you should ask, it is not exhaustive. The participants should raise their own concerns and requirements even if (especially if) they are not covered in the workshop. You may also add your own questions to this template.You can also gather the integration requirements or other technical notes about how the process should work, but only if already known.

This phase raises even more specific questions than the Analytical phase that are aimed at discovering exact details about each activity in your workflow. Let's look at an example question from section 1.3.

- What happens if the workflow errors?
- The workflow may fail for any number of reasons. (For example, a system task could not be completed and could not be handled at the task level). Should anything specific happen if the workflow fails?

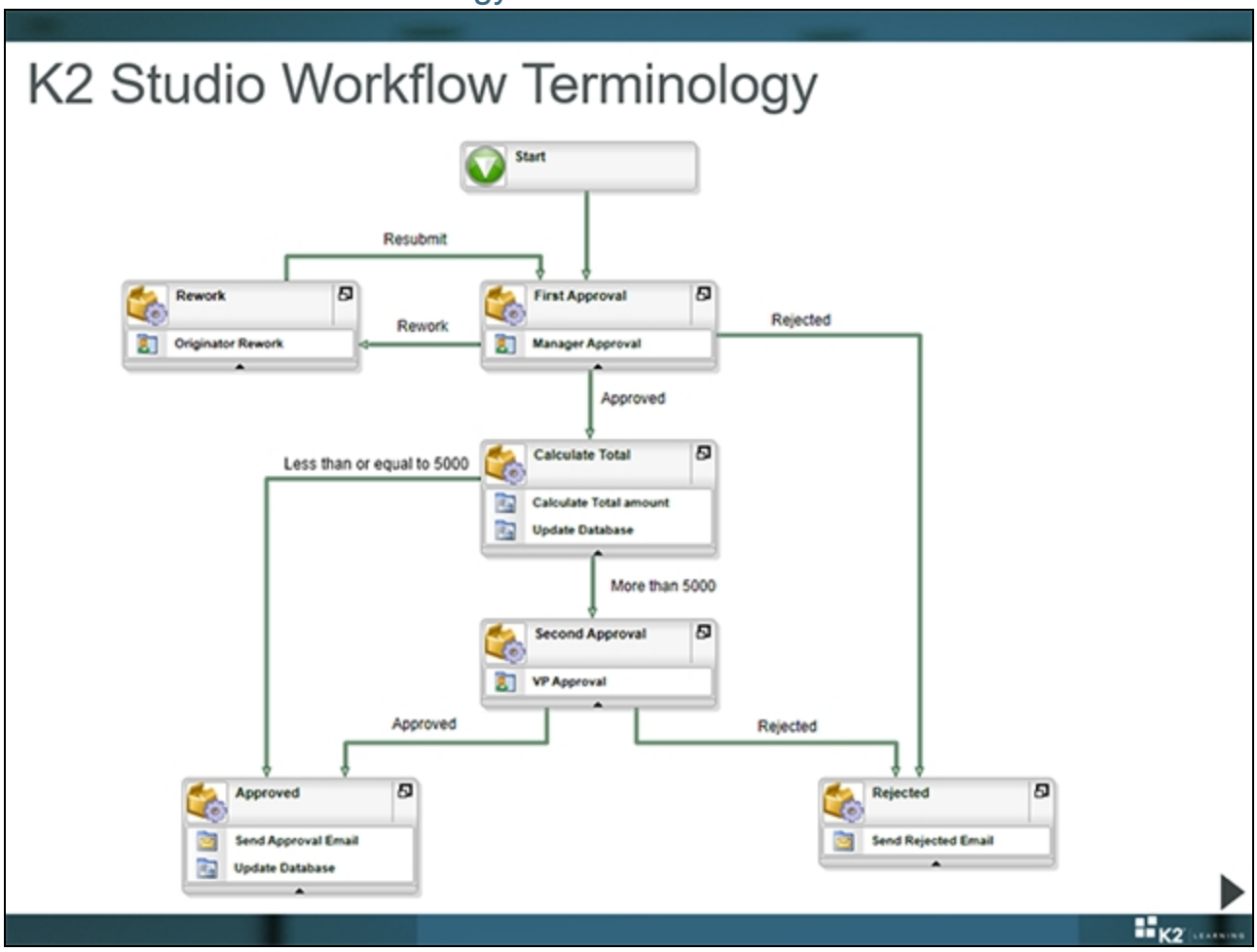
Questions like this are aimed at anticipating potential problems, identifying causes, and preparing solutions. Do not try to solve technical design or implementation questions during this phase. The purpose of this template is to guide discovery and gather information. Trying to solve architectural or technical implementation can derail the discovery phase, so leave that for the technical specification phase.

Using BPMN to Model Processes



We mentioned using BPMN to model workflows earlier. This is a good way to ensure that your process model is understandable and uses consistent notation. The image here illustrates the sample expense claim approval solution that we explore in greater depth in Interface Modeling.

K2 Studio Workflow Terminology



While you may not necessarily be implementing workflows using the K2 Design tools, it is a good idea to quickly cover the terminology used in K2 workflows so that you understand why we gather the requirements we do and why the specification document includes as much detail as it does. If the designer and the developer share a common language, it is easier for them to communicate the requirements and easier for the developer to implement the workflow.

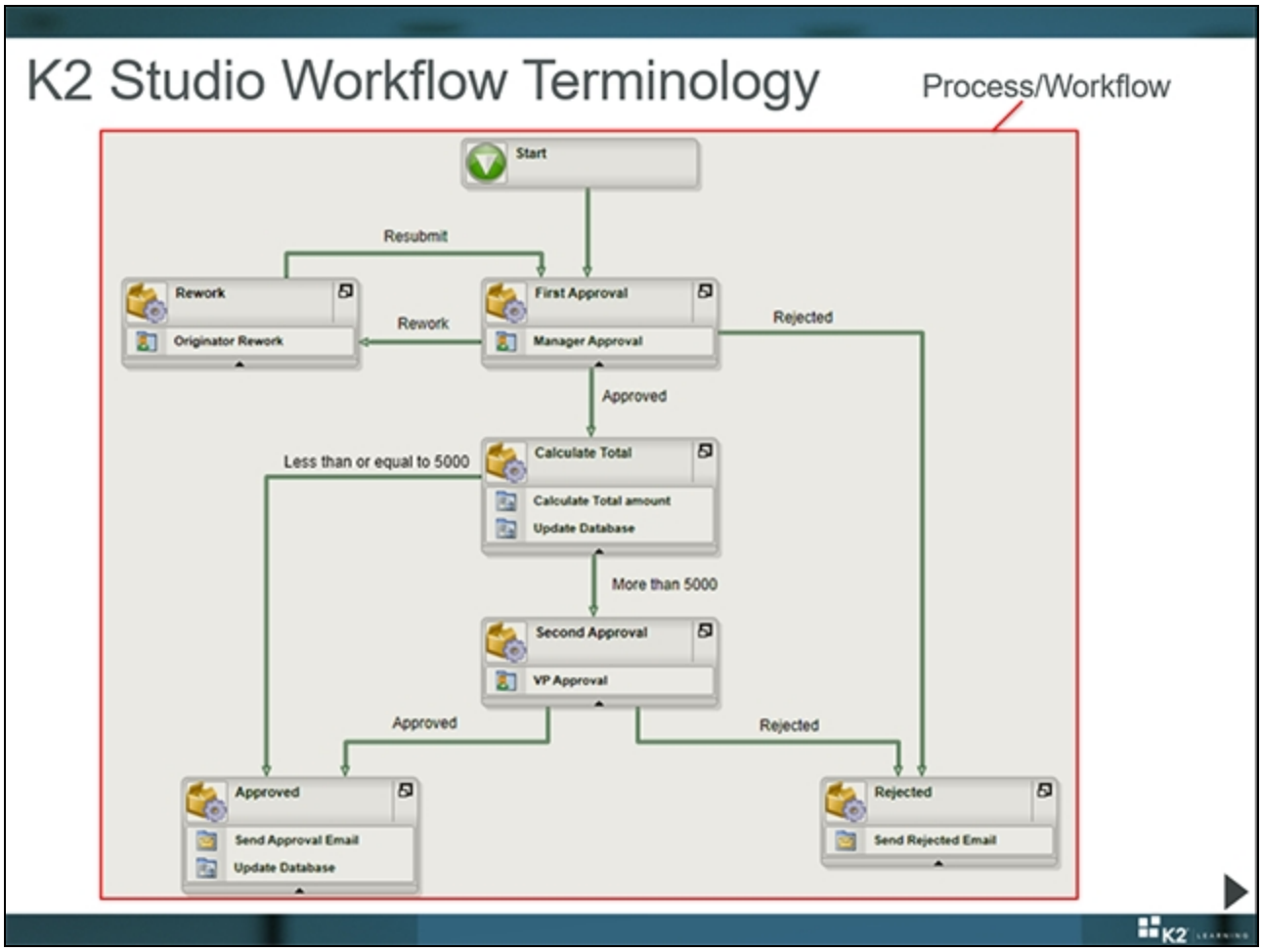
The diagram we have provided is the K2 workflow built for the sample Expense Claim Solution. Notice that the sample does not really map 1:1 with the BPMN diagram, but we will explain why this is in just a minute.

Process Workflow

A process (also known as a workflow) refers to the entire process designed in the K2 design tool. A workflow will have at least a Start activity and one other activity (otherwise it wouldn't be a workflow, right?). In K2, the terms "Process" and "Workflow" are used interchangeably, because they refer to the same thing: the entire process with all of the activities, events, lines and properties that make up the process. Similarly, the terms "Process Instance" and "Workflow Instance" also refer to the same concept: it is an instance of the workflow design.

Processes usually contain multiple activities, which are joined with lines. The activities and lines generally define the flow of the process. The events inside the activities define the work performed in the process, and the rules behind activities define how the activities are executed.

In the diagram below, all of the activities, events, lines, and the rules behind each of these items make-up the process.

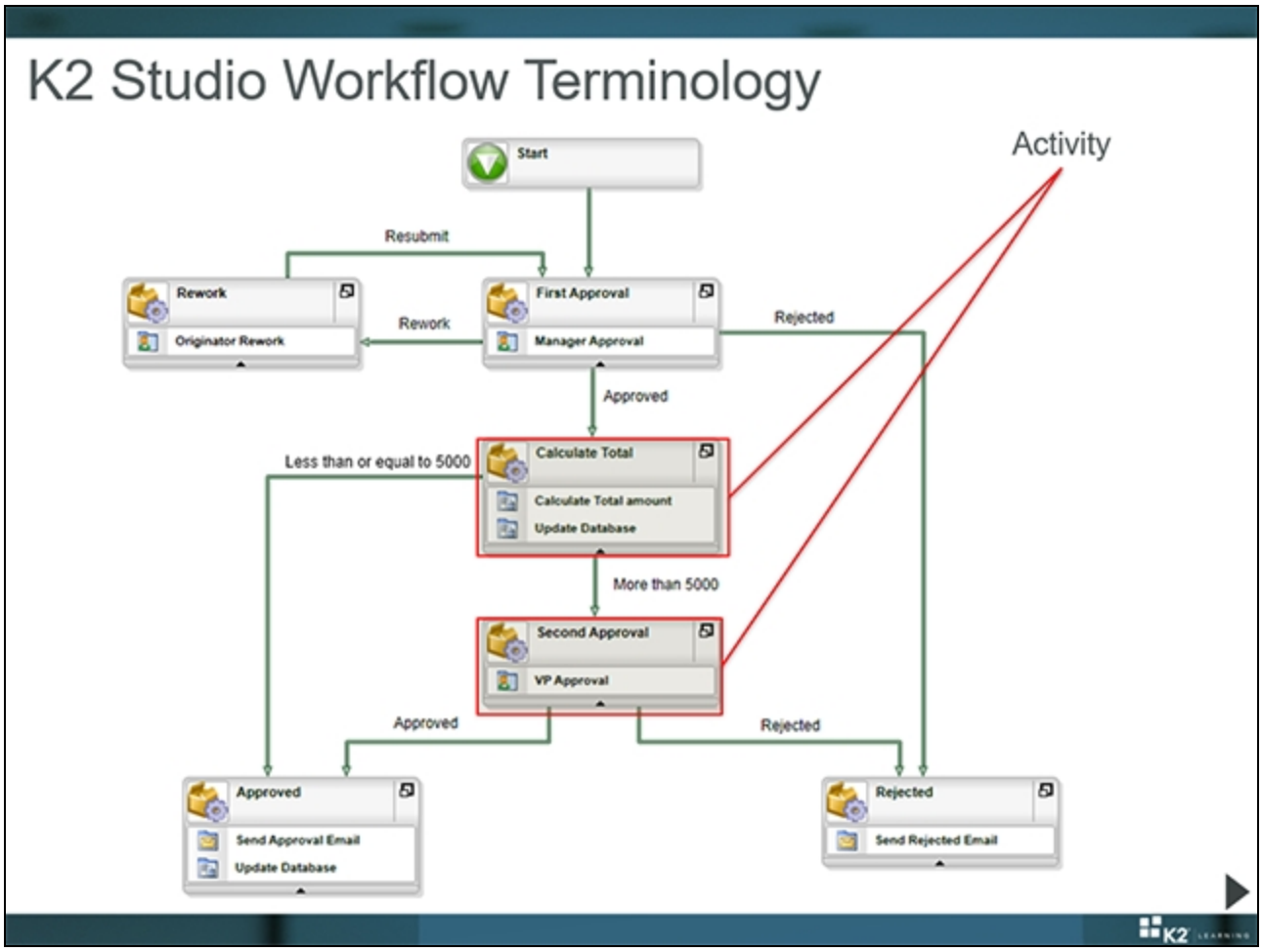


Activity

An Activity is a step in a process, and acts as a container for work that will be performed by a user or a system. Whenever work moves from one person to another or to the K2 server, a separate activity will be created for each of these roles. (Strictly speaking, it IS possible to perform both user tasks and server tasks in the same activity, but think of activities as indicators representing that work moves from one human/system to another human/system).

The actual work in the activity is represented by events in that activity. Activities must always contain at least one event but could contain more than one event.

In the diagram below, two activities are highlighted. In this case, the activity **Second Approval** is performed by a user, and the activity **Calculate Total** contains two server-side events performed by the K2 server.



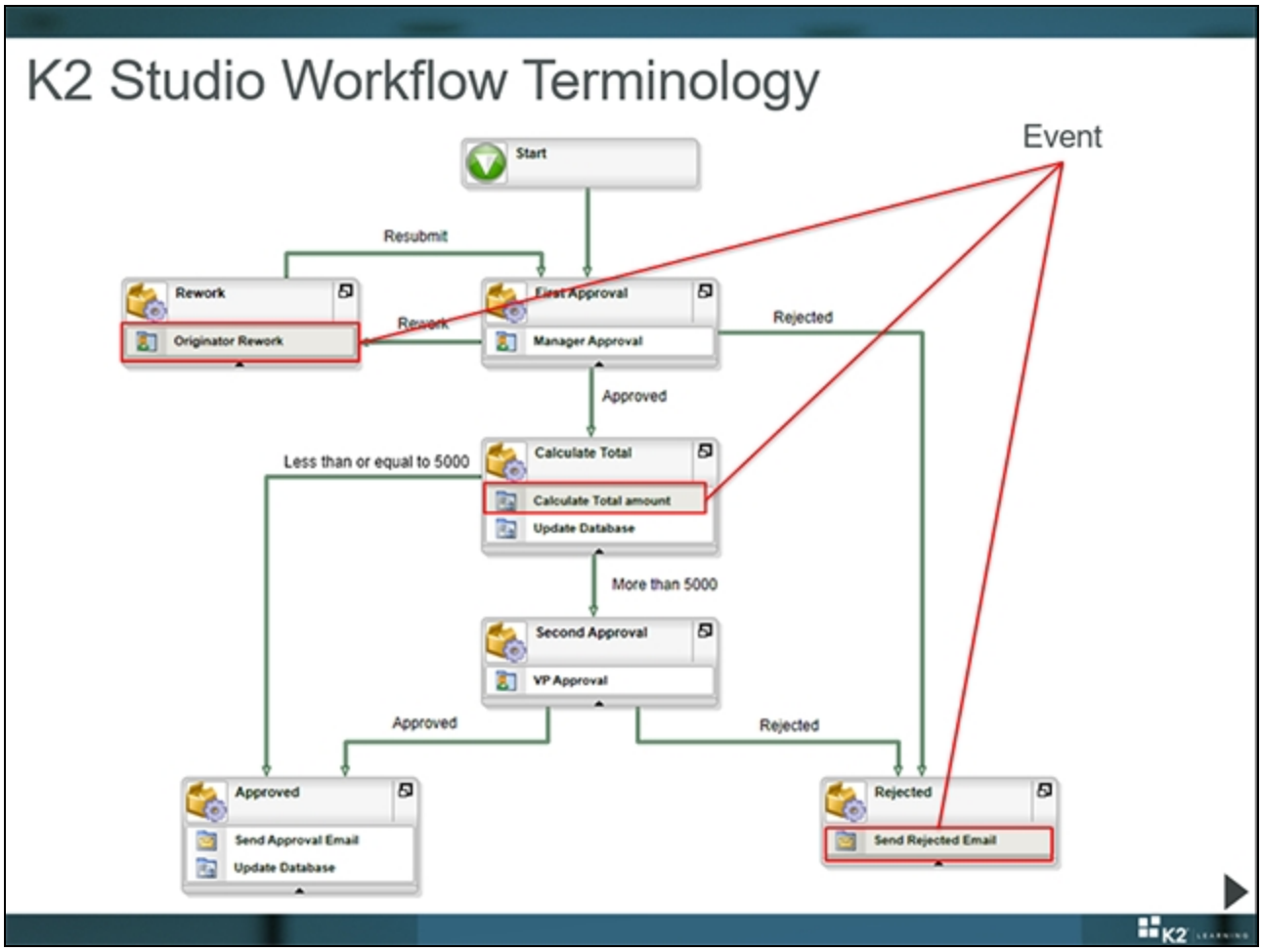
It is very common to define various rules for an activity that will control how that activity should execute at runtime. Activities may contain rules that define IF and WHEN that activity should execute, WHO will perform the task if the activity must be completed by a user and WHAT the possible outcomes for the activity could be. A process will always have at least one activity that follows from the Start activity.

Event

An event is a unit of work in a process, which is performed by a human or system. Work performed by a user is known as a Client Event and work performed by the K2 server is known as a Server Event.

Events must always be contained inside activities, and the event will describe the work that is being performed. K2 provides many wizards to make the setup of these events very easy; for example, the E-mail Event wizard that allows the designer to configure an email that will be sent as part of the process. This wizard gathers input from the designer and lets them drag and drop variables or type text into the various properties on an email message.

In the diagram below, three events are highlighted. In this case, **Originator Rework** is performed by a user, and the other Events **Calculate Total amount** and **Send Rejected Email** are performed by the K2 server.



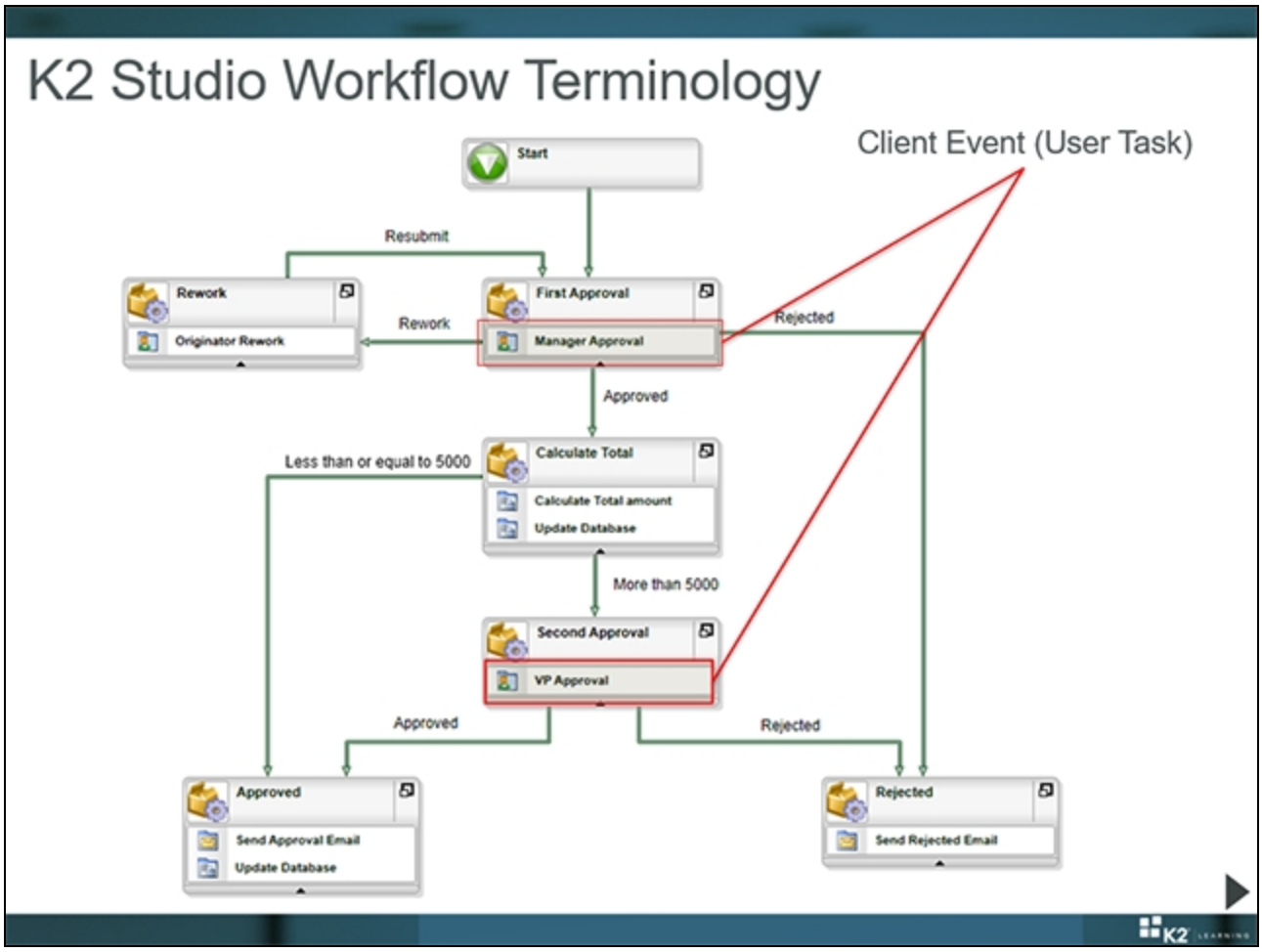
Notice that the **Calculate Total amount** and the **Approved** activities each contain two events. Activities can contain as many events as needed to complete the step, and for server-side processing it is very common to have multiple events in the same activity.

Client Event

A Client Event is a task which is performed by a human. All tasks that are performed by humans must be allocated to a destination (User or Users) which is defined on the activity level. This destination determines who will ultimately get the K2 task on their task list.

Most often, user tasks require the participant to make some decision: perhaps they need to approve or reject a request, or send it back for rework. Users do not always need to select a decision though. In some cases, the participant may only need to indicate that the task has been completed in order for the workflow to continue. The key, however, is that a person had to tell the workflow to continue. Any process that is currently at a Client Event will remain at that step until the user completes the task, or an escalation causes the task to be expired. (A process administrator may also override the process and force the workflow to go to another step. For more information, see the learning modules that deal with K2 Process Administration.)

In the diagram below, two user tasks have been highlighted: **Manager Approval** and **VP Approval**. Each of these tasks will be completed by the relevant users with one or more user interfaces.

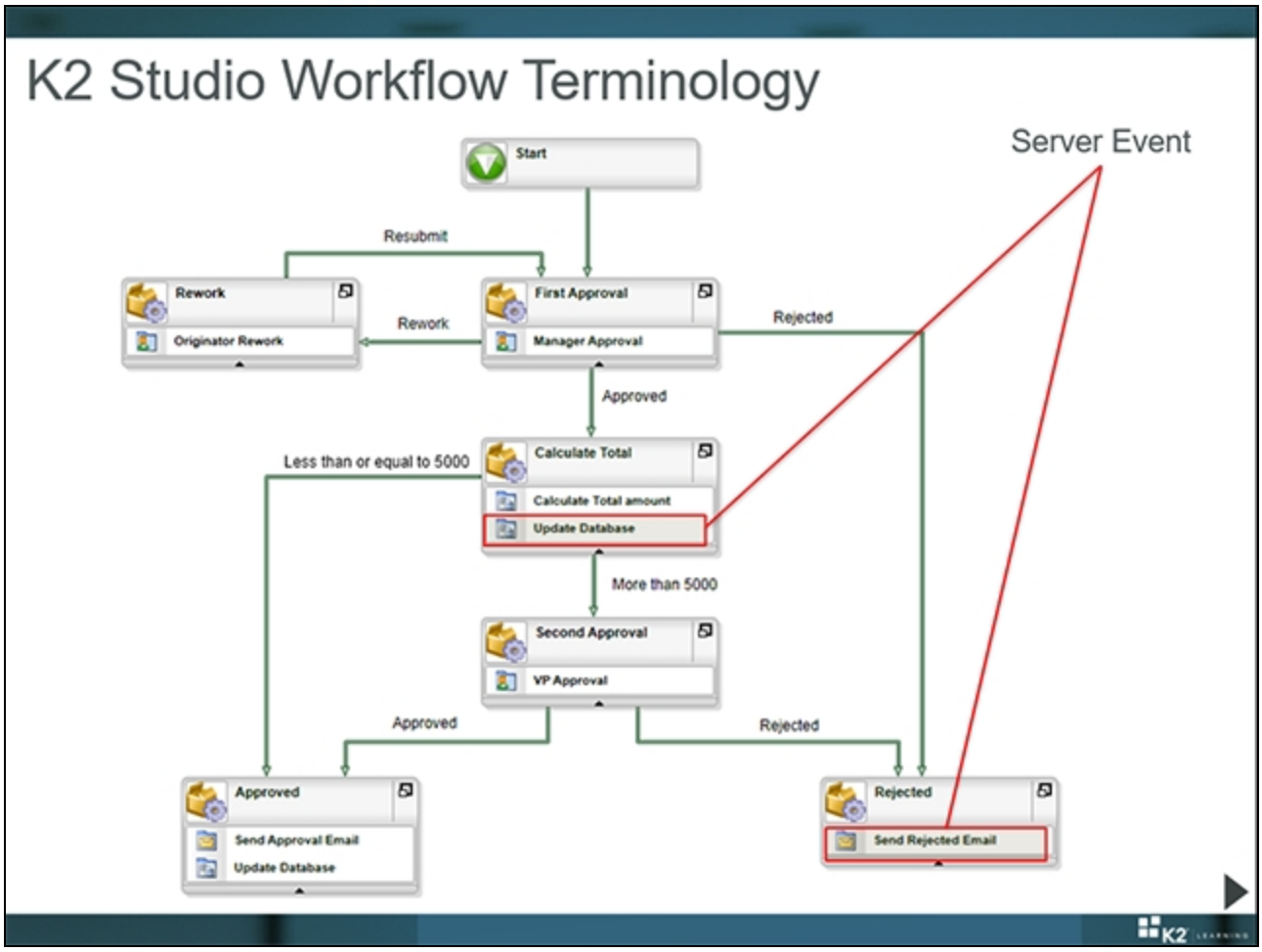


Note that all Client Events must have a destination set on their containing activity before the process can be deployed. After all, if a task is assigned to a user, K2 must know who to assign the task to. This destination could be a hard-coded username, based on a group in AD or SharePoint; a role defined in K2, or could be looked up in some external system. The K2 workflow doesn't really care HOW you determine who should do the work. It just cares that some username or group name is returned that tell K2 who to assign the work to.

Server Event

Server Events are tasks which are performed by the K2 server. These tasks are usually not long-lasting and will complete in a few seconds, at most. Server Events could involve integration with some system, and the exact nature of that interaction depends on the wizard or SmartObject method being executed in the event. Server Events are also used to perform some server-side processing such as performing calculations or manipulating data.

In the diagram below, two Server Events have been highlighted: in the **Update Database** event, K2 is using some code to interact with a database, and in the **Send Rejected Email** event K2 is generating and sending an email.

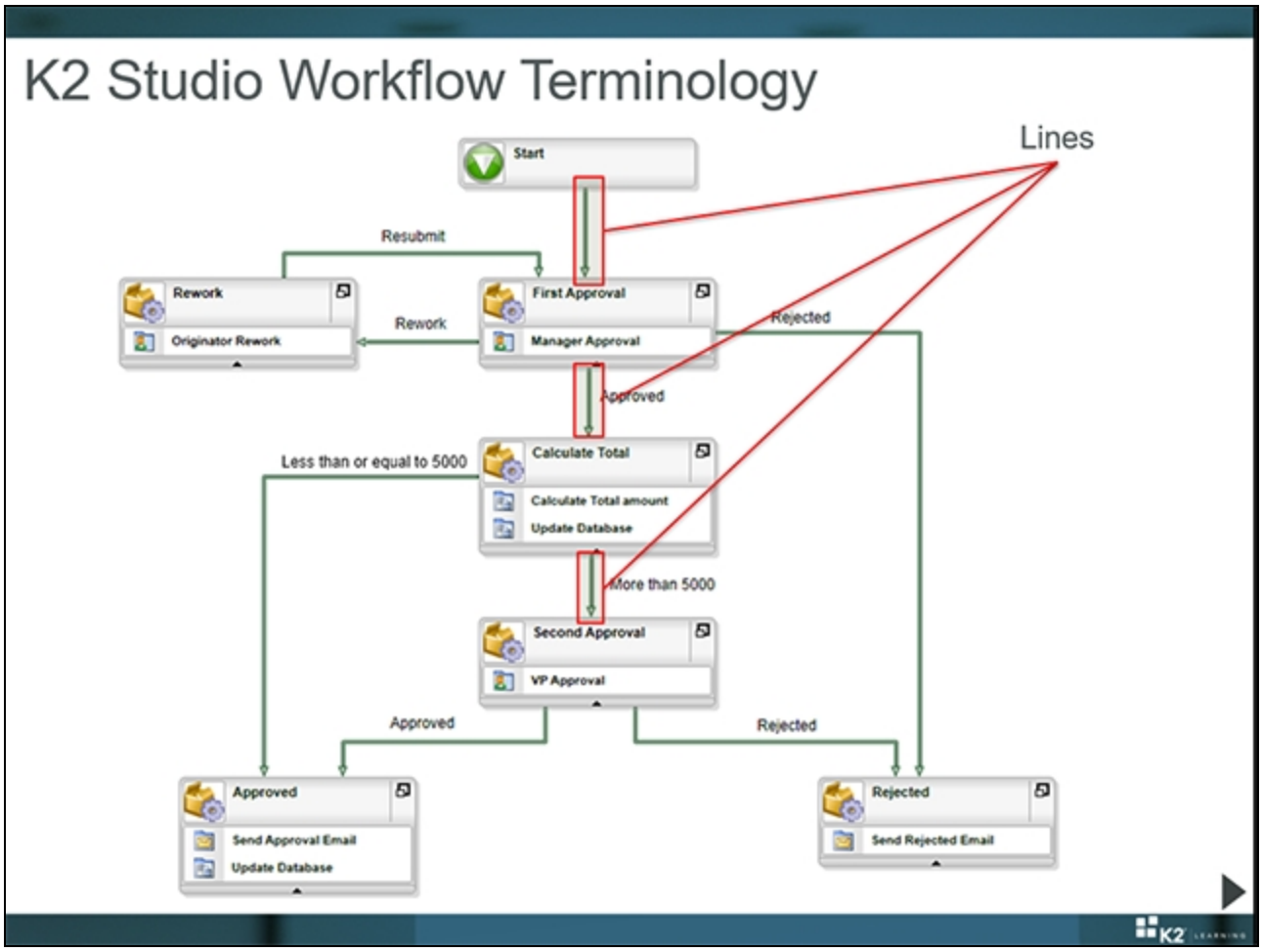


Lines

Lines are the possible paths in a workflow. Lines are used to join activities in a specific sequence, and effectively tell K2 how to move from one activity to another.

Lines control the “flow” of a process, and by using Line Rules or Outcomes, workflow designers can tell K2 to follow a certain path in the process based on some condition. Lines can even be used to split and merge a process if parallel work is required.

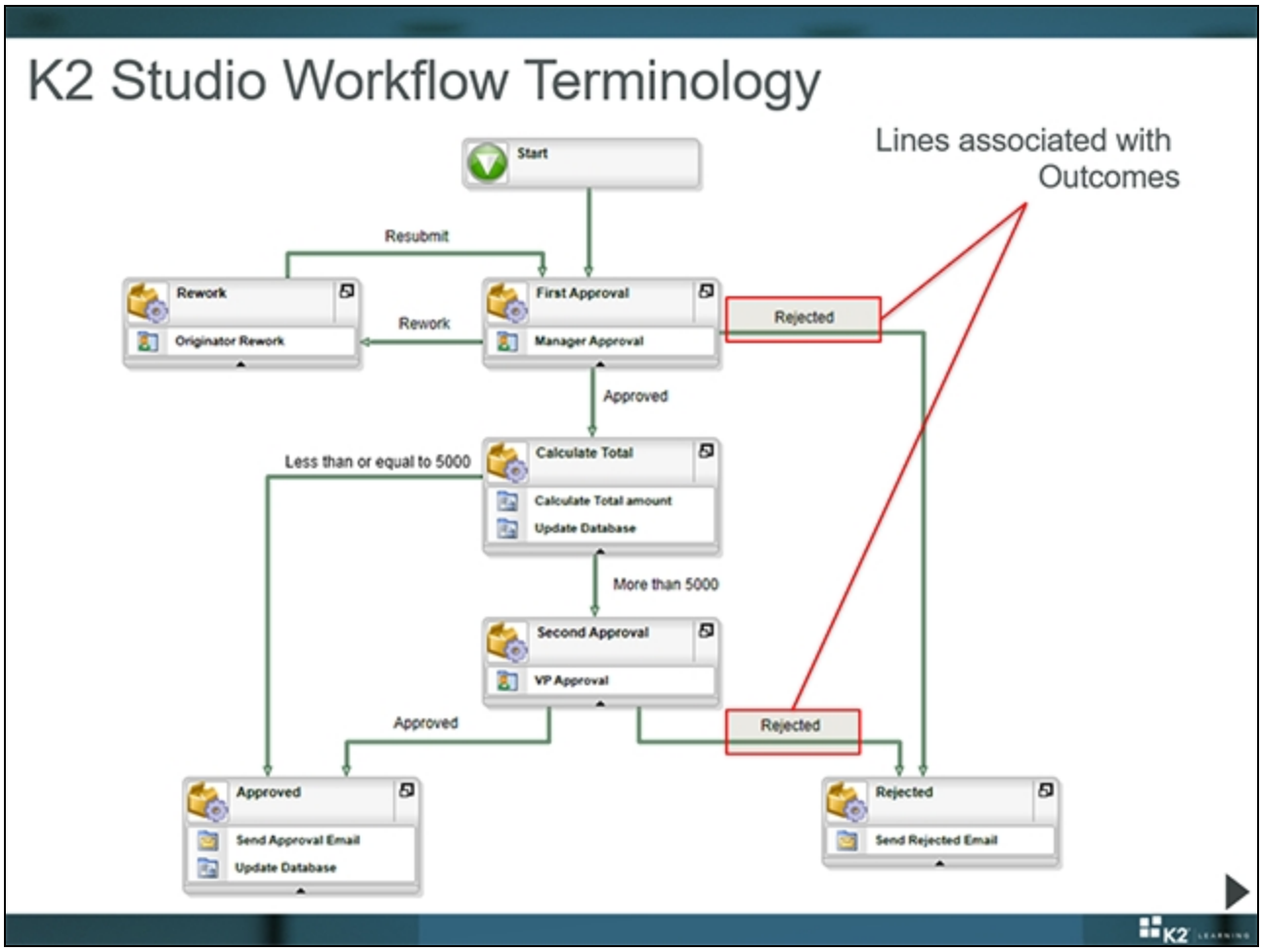
In the diagram below, three different types of lines have been highlighted. The first line just joins **Start** to the **First Approval** activity, and will always be followed since there are no rules defined for that line. The **Approved** line is a special type of line called an Outcome, and is evaluating the decision made by the approver. The **More than 5000** is a Line with a Line Rule, which is evaluating a data value in order to make a decision about where to go next. In this example, if the amount is more than 5000 the workflow must go to the **Second Approval**



Outcomes

Outcomes are a special type of Succeeding Rule and usually surface on a workflow as lines. Outcomes are usually based on one or more user's decisions (Actions) or by evaluating user actions along with other values to establish a path to follow in the process.

In the diagram below, two lines that follow certain outcomes have been highlighted. The first **Rejected** line is one of three possible outcomes resulting from the action taken in the **Manager Approval** event, while the second **Rejected** line is one of two possible outcomes resulting from the action taken in the **VP Approval** client event.

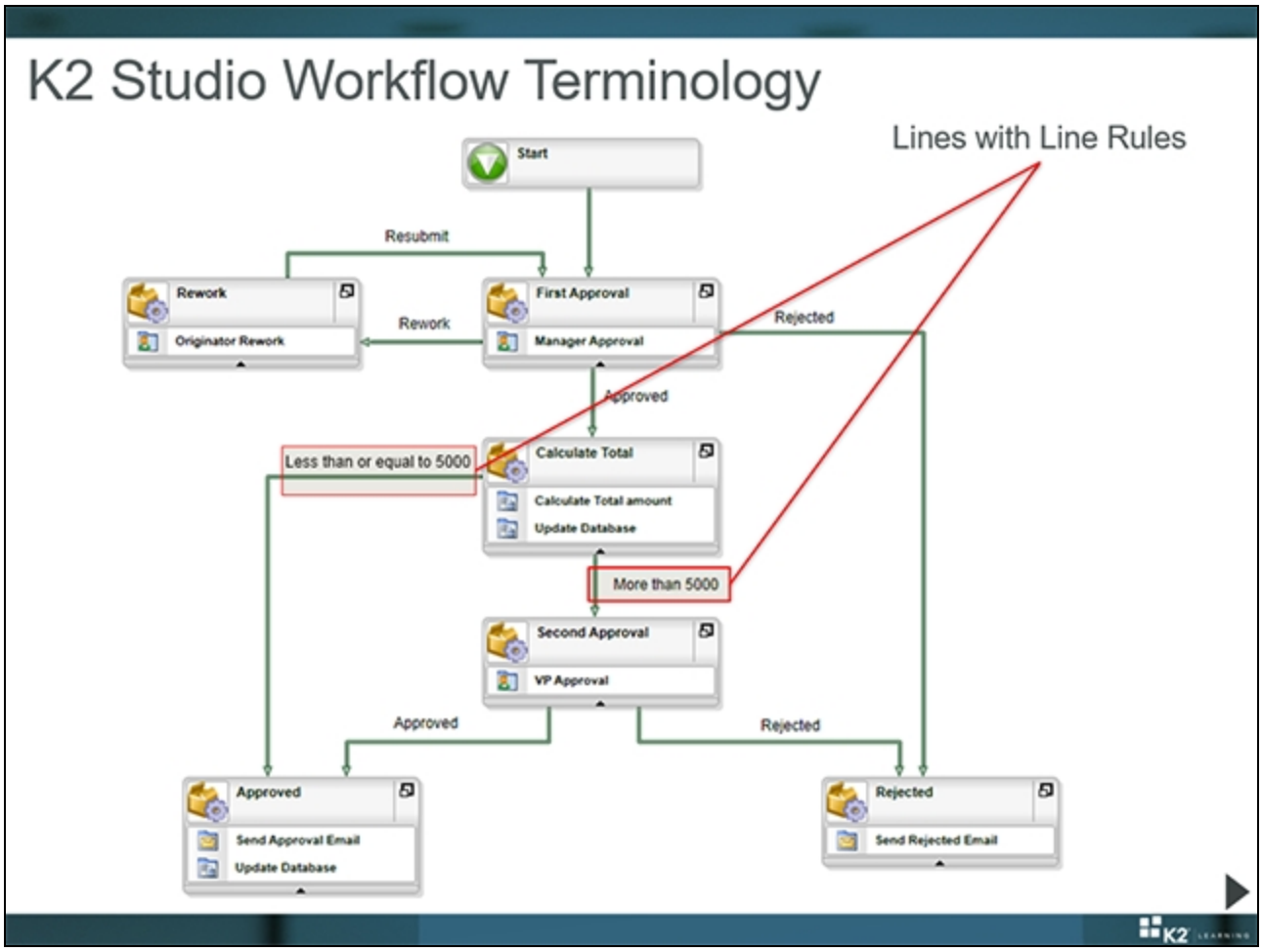


Behind the scenes, outcomes are defined as Succeeding Rules for workflow activities, and the resulting outcome is saved to an activity-level datafield. The lines that flow from the activity evaluate this activity datafield. Even though we have indicated "Outcomes" on the diagram below as lines, strictly speaking, outcomes are actually defined on an activity level.

Of course, outcomes and lines can be a lot more complex than the example above, such as multiple parallel lines, outcomes that evaluate both user input and data values, and so on. For the purposes of this discussion, just remember that in most cases, actions yield outcomes which yield lines.

Lines with Line Rules

Workflow designers can define additional Line Rules behind lines to control the flow of a process. Most often this is used to control the path of a process based on the result of some comparison operation. In the diagram below, two lines with Line Rules have been highlighted – each line checks the Total value of the request. If the value is less than or equal to 5000, the process goes directly to the **Approved** activity, but if the value is more than 5000, the process goes to the **Second Approval** activity.



Escalations

Escalations are tasks performed by the K2 server in response to some delay time being exceeded. Escalations can be configured for events, activities, and the process as a whole, and will only fire if that event, activity, or process has not completed within the specified interval.

Events, activities, and the process could have several escalations that can be repeated as many times as desired. This makes it possible to design escalating levels of escalations on any level.

An example of an escalation is to send an email to the originator of the process if the first approver has not completed the task within two days of the process being started. You could add another escalation to the first approval activity to forward the task to another user if the original approver hasn't completed the task after 3 days.

Notifications

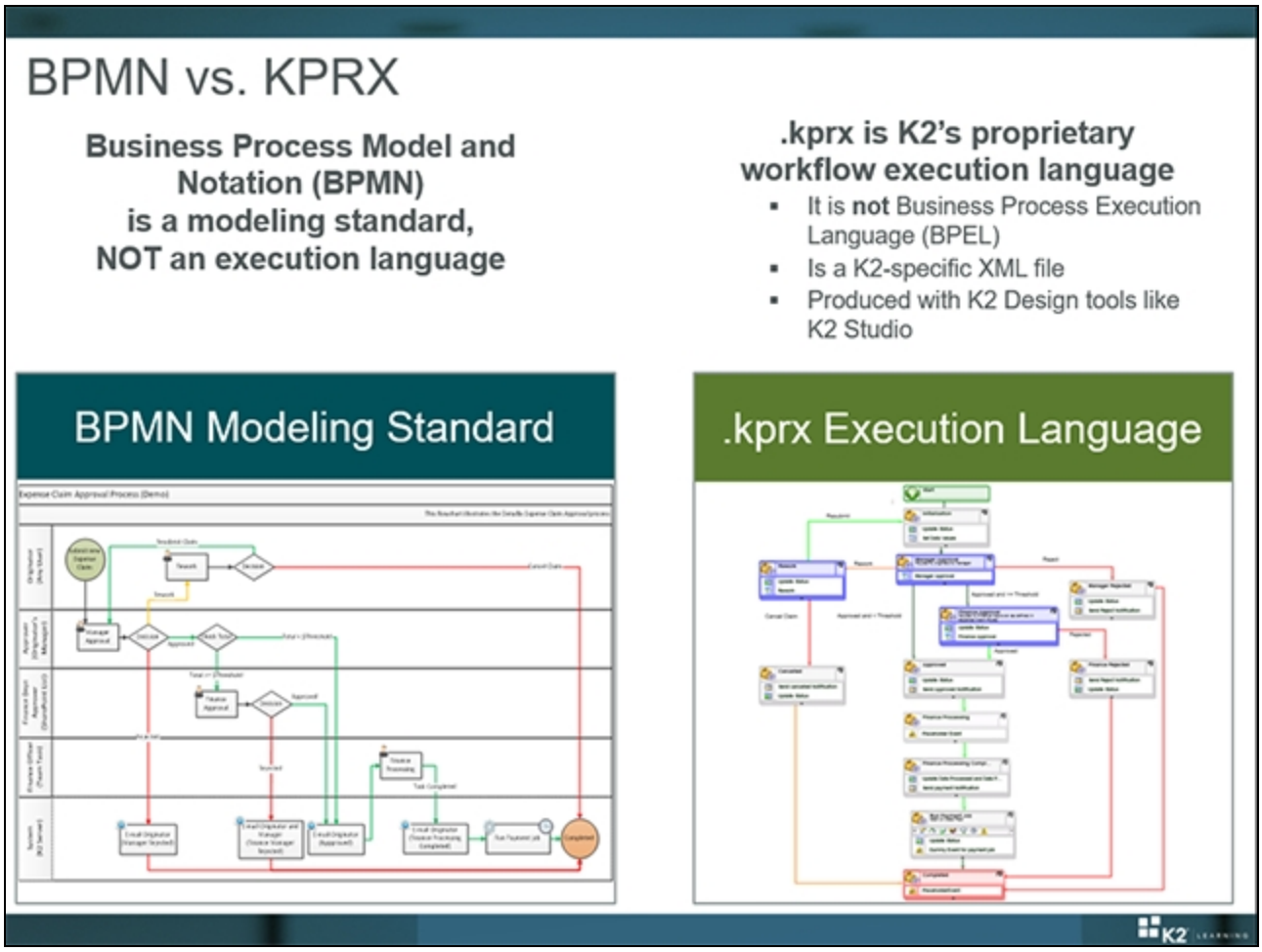
Notifications are emails that are sent to users during process execution. Notifications are typically used to alert participants that they have new tasks, and are used in escalations as well. Both the styling (for example, applying custom headers and footers or fonts) and content (the message itself and the data included in the message) of the notification can be customized. SmartActions enable users to complete their tasks just by replying to the task notification email that was sent to them. The workflow designer can decide whether to allow users to complete tasks this way, or whether they must always open the form to review the task before they can complete the task.

Summary

- Process/Workflow refers to the activities, events, lines, and properties that make up the process
- Activities are "steps" in the workflow joined by lines, and contain one or more events
- Events are the tasks/work that get done in an activity, and could be Client Events (human tasks) or Server Events (system tasks)

- Lines connect activities and define the flow of a workflow
  - Lines can be associated with outcomes (e.g. the result of the decisions/actions taken in a client event).
  - Lines can contain Line Rules that perform some logic to determine workflow routing

BPMN vs. KPRX



BPMN and .KPRX are not only NOT the same, they are **very different**. So why the distinction? While the BPMN specification provides a mapping between the graphics of the notation and the underlying constructs of execution languages, that connection can be enough to cause some degree of confusion.

Business Process Model and Notation (BPMN) provides businesses with the capability of understanding their internal business procedures in a graphical notation that gives organizations the ability to communicate these procedures in a standard manner.

Therefore, in brief BPMN is:

- A graphical notation
- A standard for process modeling
- A standard for visually communicating business processes
- A type of flowcharting based on Unified Modeling Language

.KPRX on the other hand is actual code that K2's workflow engine executes. When designers build K2 workflows with the K2 workflow design tools, they are building KPRX files, which are then deployed to a K2 server for execution.

WORKSHOP SESSION 2: PROCESS MODELING

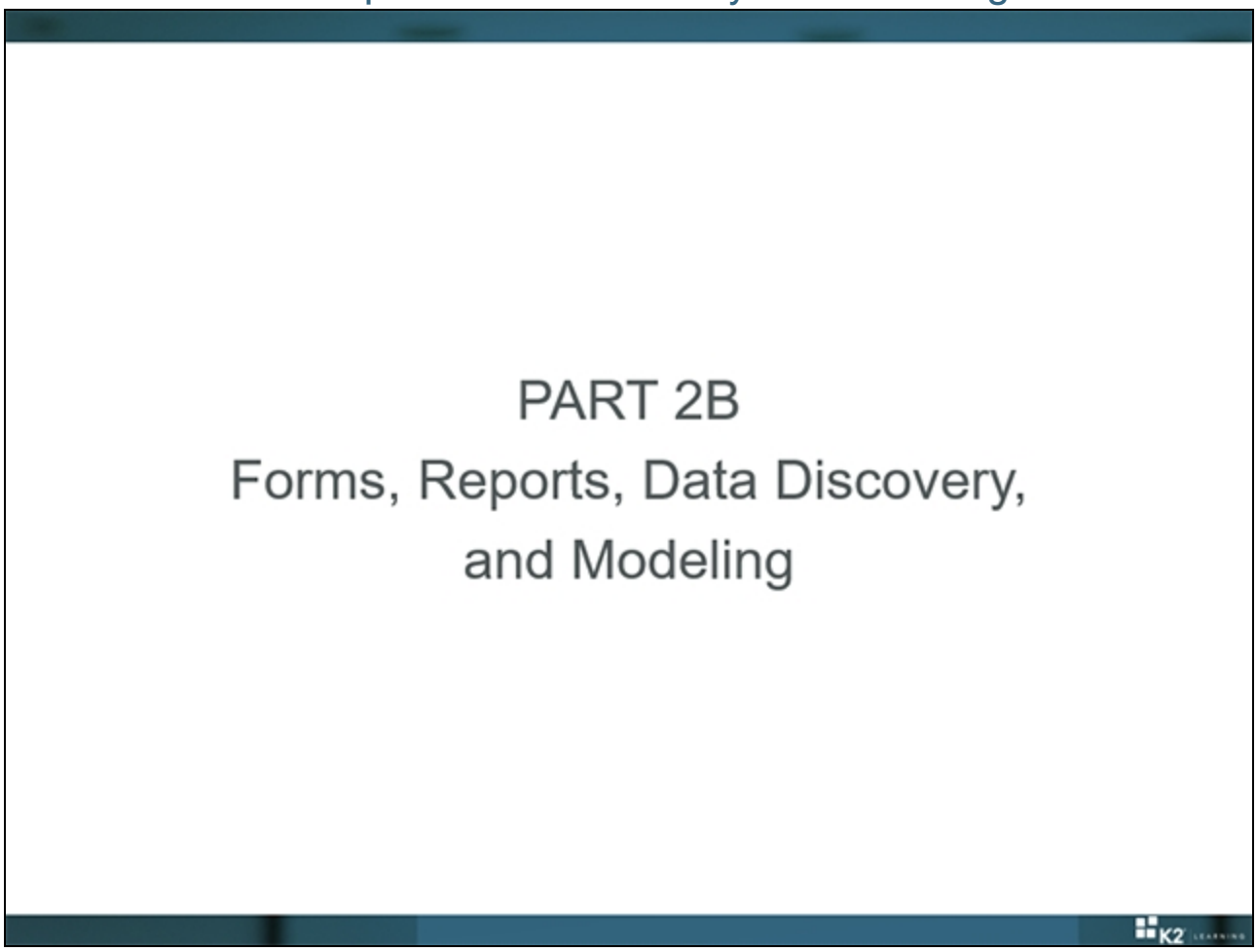
Note
The workshop exercises in this learning module use several resources, for example template documents and workshop guides. If you have not done so already, please download the following file from K2's help site: http://help.k2.com/files/10701.
Once the zip file is downloaded, please extract the contents of the zip file to your computer. (Alternatively, you can also extract the zip file to the virtual K2 environment provisioned for you for this training course.)

In this workshop we will take the hypothetical expense claim approval workflow and put it through the process Descriptive - Analytical - Specification discovery steps. Let's use some template documents and put the hypothetical expense claim approval requirements through them. We can use the completed document templates to help guide the conversations.

After downloading and extracting the resources for this module as described in the note above, you can find the sample documents for this workshop at the following location:
[Extracted Location]\Workshop Resources\Workshop Session 2 - Process Modelling

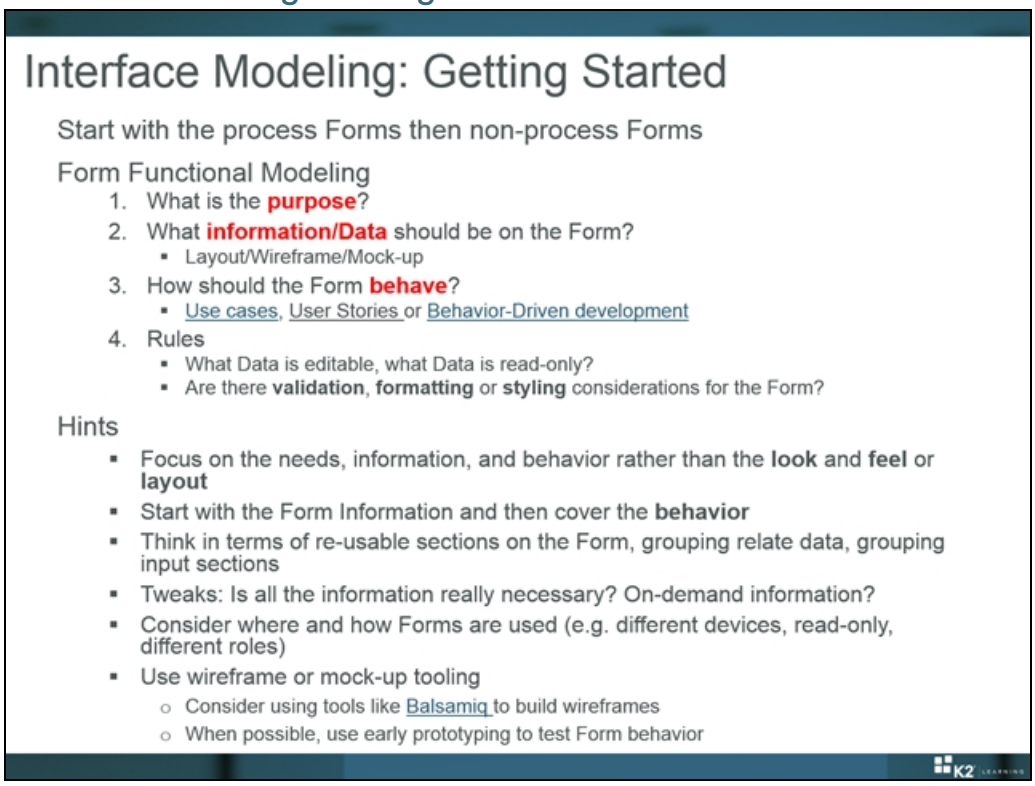
Use the **5. K2.Analysis and Planning.Discovery and Modelling Checklist Template.docx** document to guide the questions to ask and answer in this workshop.

PART 2B Forms Reports Data Discovery and Modeling



In the this part we will look at discovering and modeling Forms, Reports, and Data for K2 applications.

Interface Modeling: Getting Started

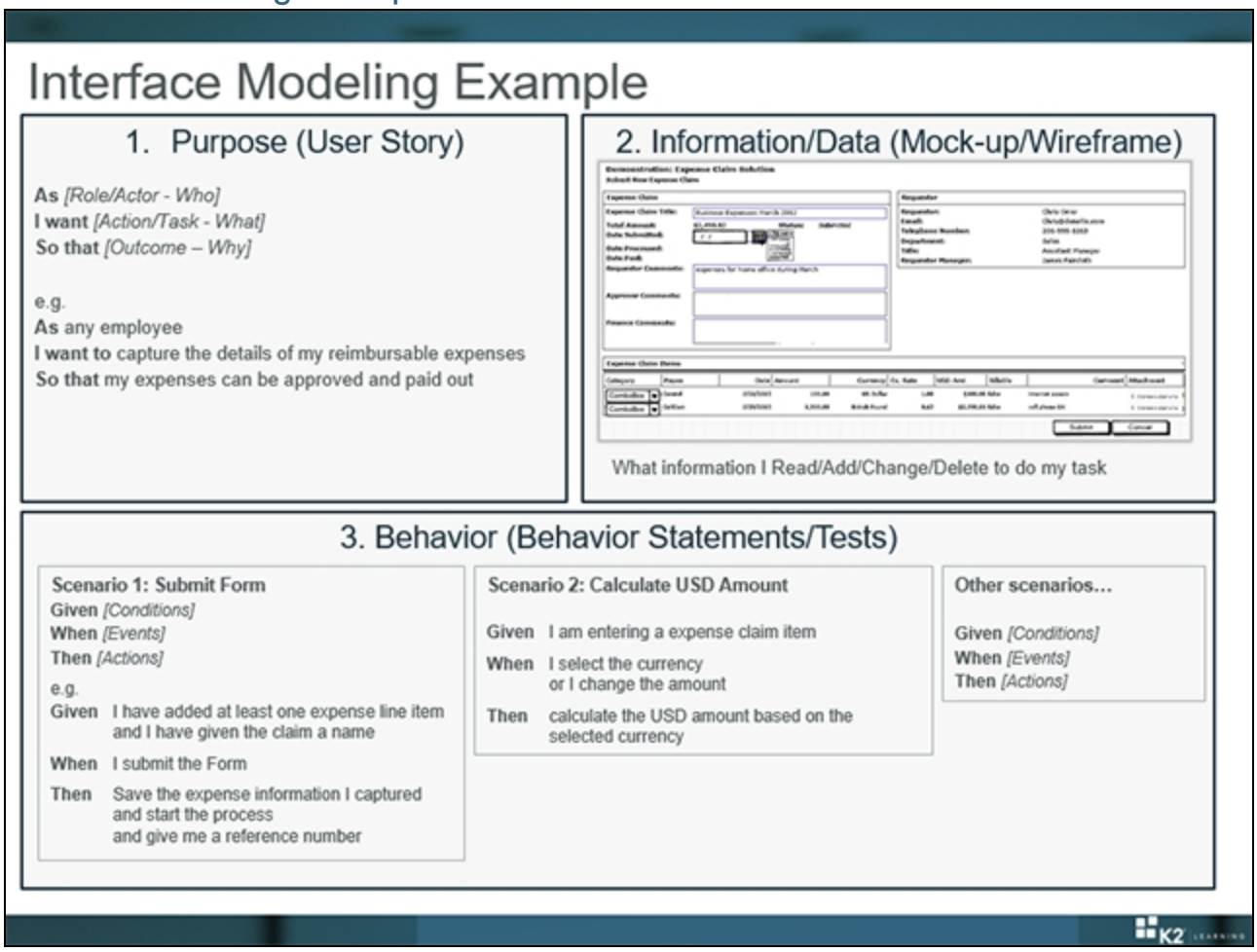


The first place to start when modeling Forms is with the process/workflow model. You should just have completed the process modeling ("descriptive" level at least, so now you have an idea of the user tasks in the workflow). Now you can start to discover and gather requirements for the user interfaces.

When doing discovery and functional modeling, always bear in mind the question, "What is the purpose of this form?" It is easy to get bogged down in the layout or controls. If this happens pull the conversation back to the purpose of the form, the information on the form (NOT necessarily the data, but the information that the user needs to see and modify) and the behavior of the form. Then go into details about the form data and validation, expressions, and formatting.

The hints suggest some ways you can keep on track when modeling interfaces. We recommend using modeling tools like Balsamiq or even sticky notes to model out what the form should look like on a high-level. Again try not to get too bogged down. The information, behavior, and purpose are most important.

Interface Modeling Example



This slide shows an example of how you can model interfaces. You can use this model "as-is" or design your own. Regardless, this model shows an "inventory" of what should be captured or explained in your model.

Let's begin with the box marked, "**1. Purpose**". As the name implies, this is where you should gather the information about:

- Who is going to use your smartform
- What it will do
- What the user will expect from it

These are the essentials. You may want to add more to this section as time allows.

In the next box marked, "**2. Information/Data**" you can add a mock-up that shows:

- The form layout
- Labeling and formatting
- User interface elements like fields, drop-down menus, and buttons

Finally, in the box marked, "**3. Behavior**" define the behavior of the form using scenarios. Start with the most frequently used, then add more as needed. Notice that the format of this example records:

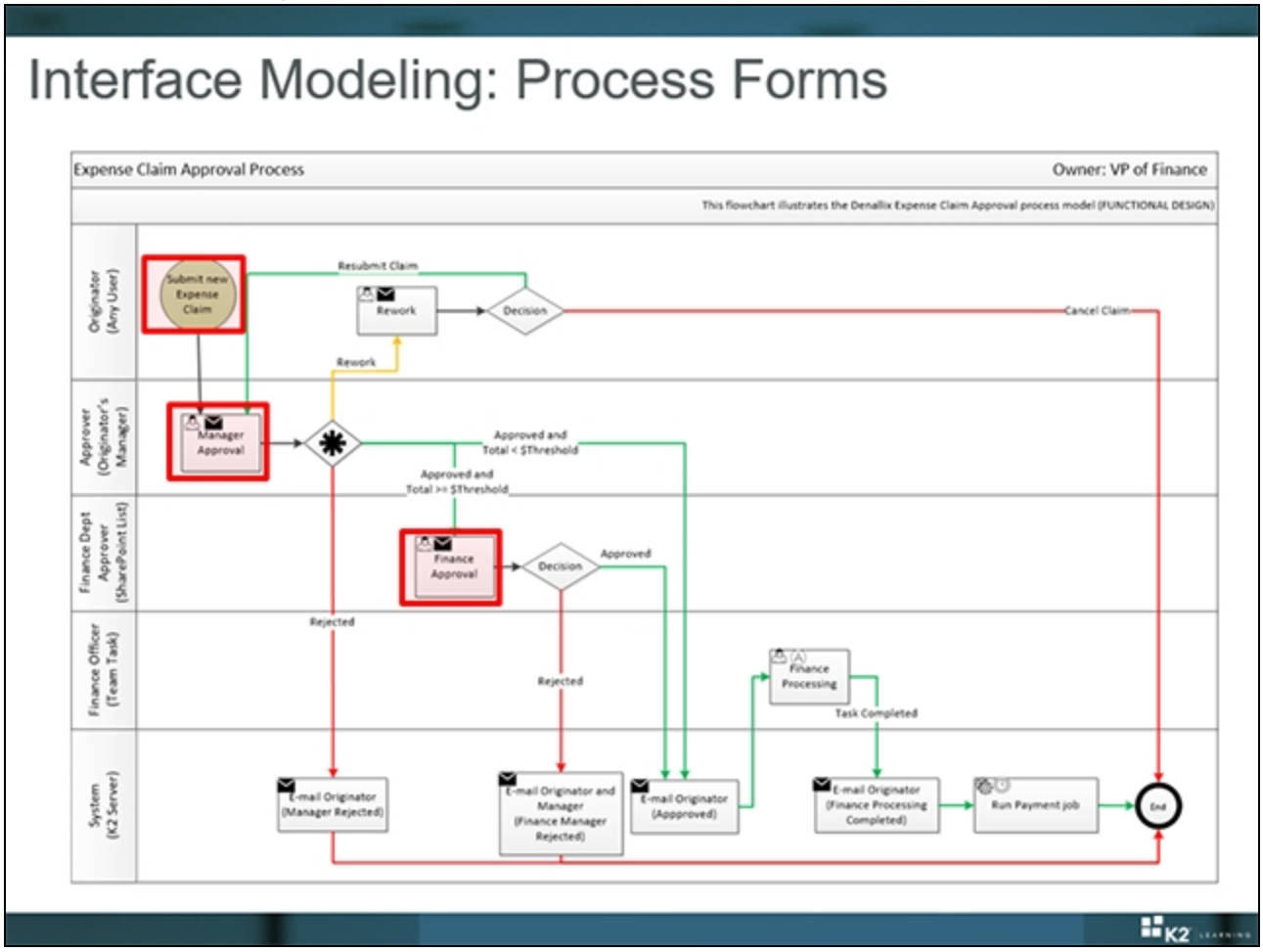
- The scenario name
- "Given" - what need or conditions stimulates the need for the form
- "Events" - what the user does as a result
- "Then" - what happens as a result

Use the KISMIM (keep it simple, make it meaningful) rule as you build out each scenario.

By sticking with these principles you can make it easier for a designer to create forms that show users the information they need to do their work. Using the syntax for specifying behavior as we did here, you ensure that the language is understandable to the business as well as the designer/developer.

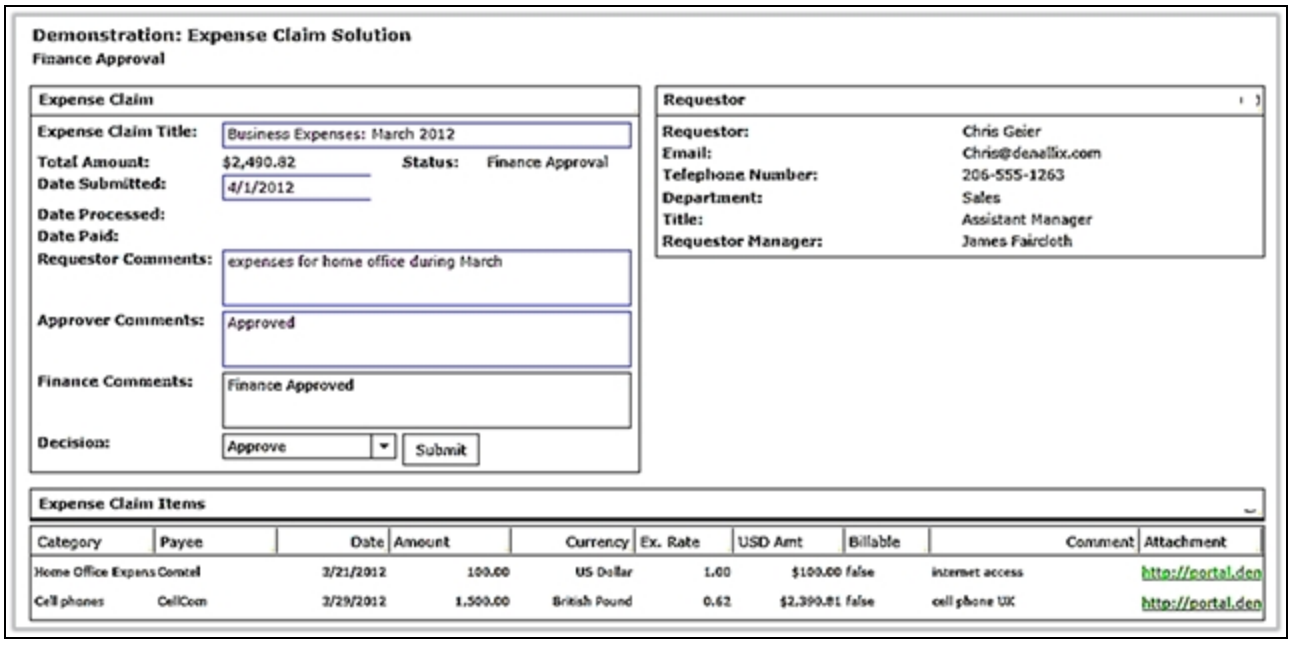
You don’t need to follow this approach just for smartforms. It works equally well when developers create custom forms in code. The point is to keep to the requirements as opposed to the implementation.

Interface Modeling: Process Forms



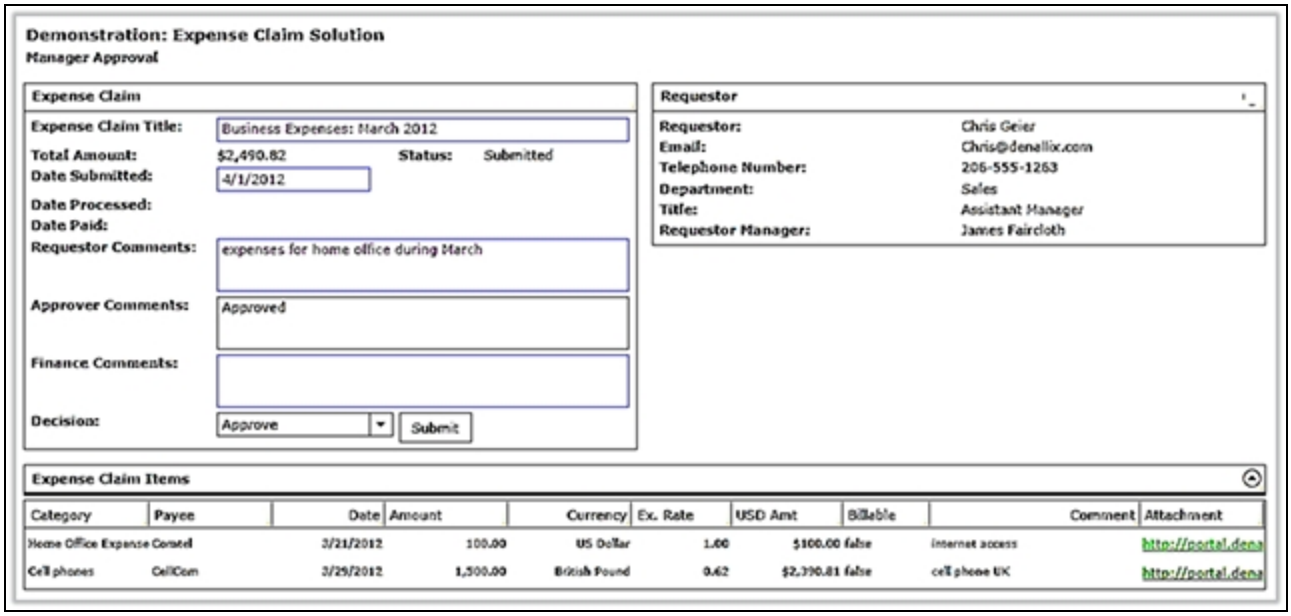
This slide is an example of UI mock-ups for the process steps in our sample expense claim approval scenario. Let's take a look at each of the highlighted boxes above.

Submit New Expense Claim



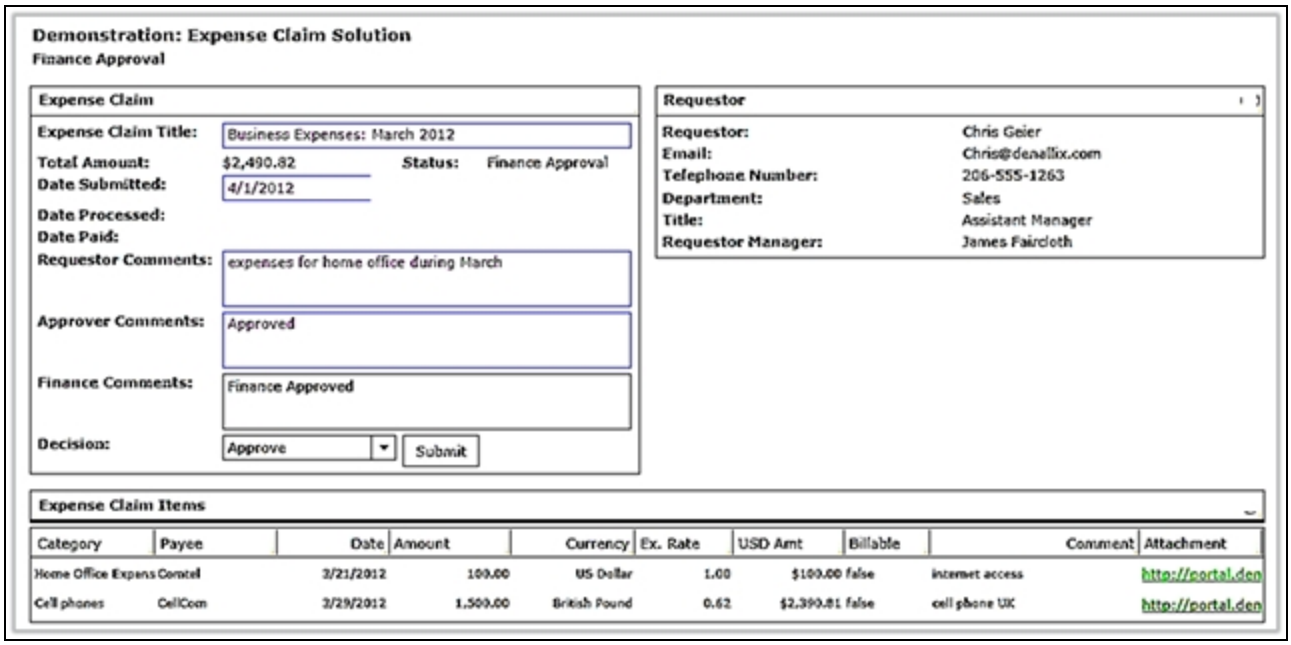
The diagram above shows how the Submit New Expense Claim form initiates the workflow. In this example, the expense claim requires manager approval.

Manager Approval



Refer to the diagram above. If the claim is rejected by the manager, the originator receives an e-mail message and the workflow ends. Otherwise, if the claim exceeds a set dollar figure, it moves on to Finance approval.

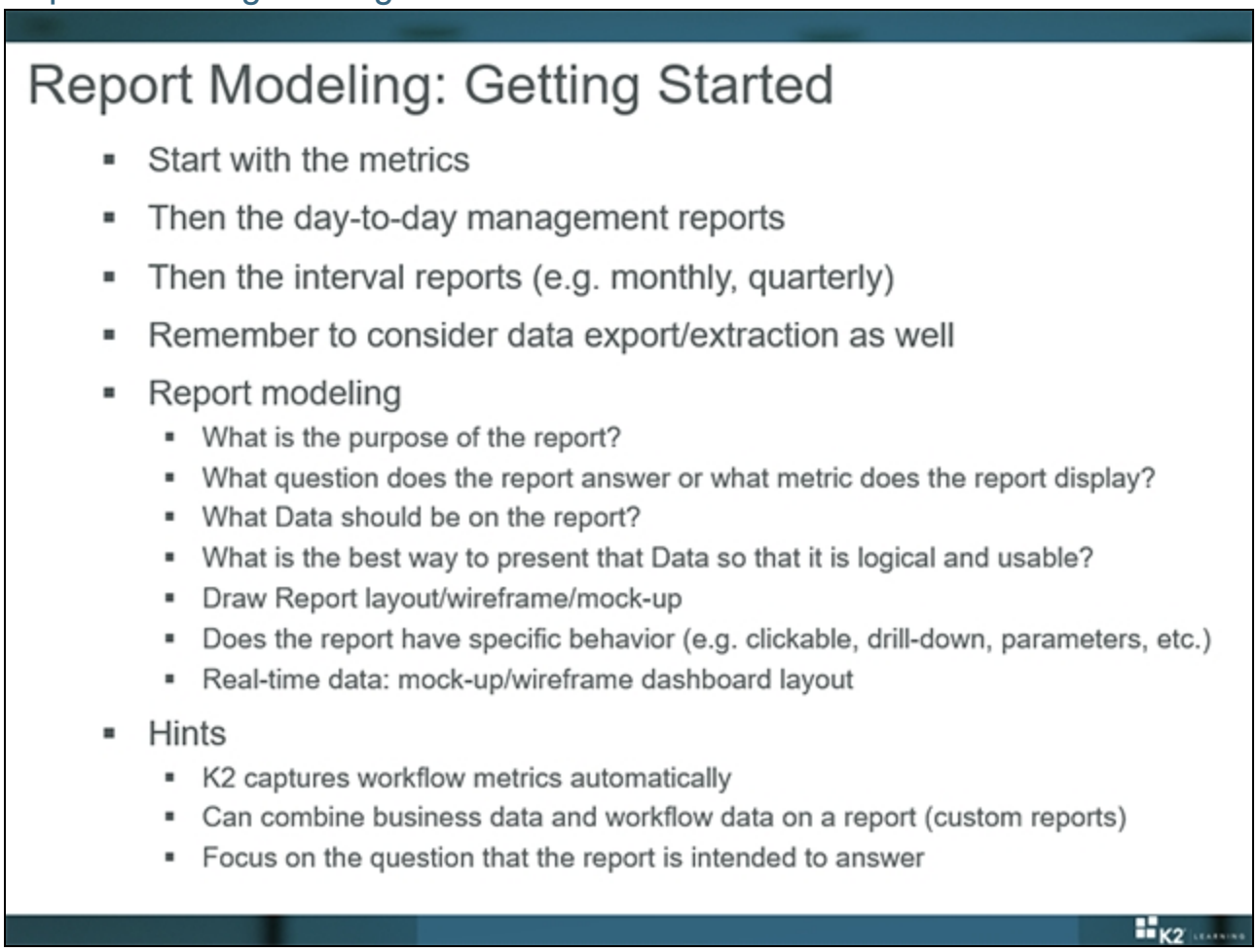
Finance Approval



Notice in the decision tree of the diagram above, that whether Finance approves or rejects the claim, the system sends an e-mail message.

If it's rejected, both the originator and manager receive a message. If it's approved, the originator receives a message, Finance processes the claim, and another message is sent to the originator when the processing is complete. Then after the payment job runs, the workflow is complete.

Report Modeling: Getting Started



Now we are ready to move on to report modeling. This slide provides a checklist or helpful framework for starting to model your report(s). This is similar to Form modeling in that you always want to keep the purpose of the report in mind.

Start with metrics. Choose the metrics that answer the question at the heart of the report. What kind of a report is it? If it's for top management, for example, that decision will influence your choices about level of metrics detail. Try to be comprehensive when choosing metrics for your report. You can always leave some out.

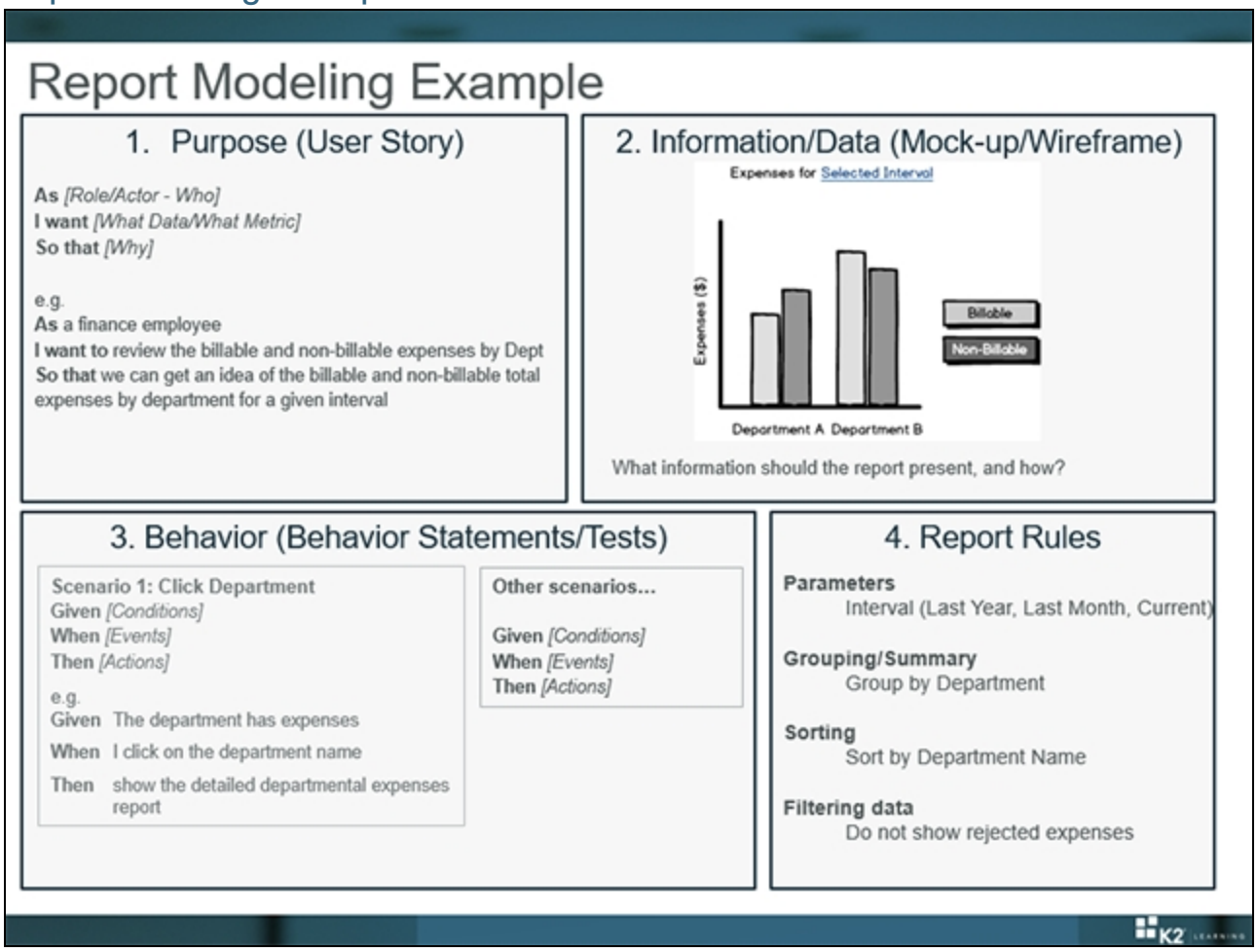
Then define the difference between daily, monthly, and quarterly reports. Next consider where the data may originate and where the finished report may be used when exported. Does it need to end up in an Excel spreadsheet for further manipulation, for example.

Report modeling from this point on involves a series of questions:

- What is the purpose of the report? Here are some examples:
  - Control costs
  - Monitor "big-ticket" expenses
  - Manage variances
- What question does the report answer or what metric does the report display? Here are some examples:
  - Expenses by project
  - Expenses by department
  - Expenses driven by an initiative
- What data should be on the report? Here are some examples:
  - Accommodation expenses
  - Lodging expenses
  - Car rental
  - Client entertainment
  - Furniture and office equipment

- What is the best way to present that data so that it is logical and usable? Here are some examples:
  - Pie charts
  - Bar graphs
  - Tables
- Draw report layout/wireframe/mock-up
- Does the report have specific behavior? Here are some examples:
  - Clickable
  - Drill-down
  - Parameters

Report Modeling Example



This slide shows an example of how you can model Reports. You can use this model "as-is" or design your own. Regardless, this model shows an "inventory" of what should be captured or explained in your model.

Let's begin with the box marked, "**1. Purpose**". As the name implies, this is where you should gather the information about:

- Who is going to use your Report
- What it will do
- What the user will expect from it

These are the essentials. You may want to add more to this section as time allows.

In the next box marked, "**2. Information/Data**" you can add a mock-up that shows:

- The Report layout
- Types and style of chart, diagram, or table
- Labeling and formatting
- Legends
- Colors

In the box marked, "**3. Behavior**" define the behavior of the Report using scenarios. Start with the most frequently used, then add more as needed. Notice that the format of this example records:

- The scenario name
- "Given" - what need or conditions stimulates the need for the Report
- "Events" - what the user does as a result
- "Then" - what happens as a result

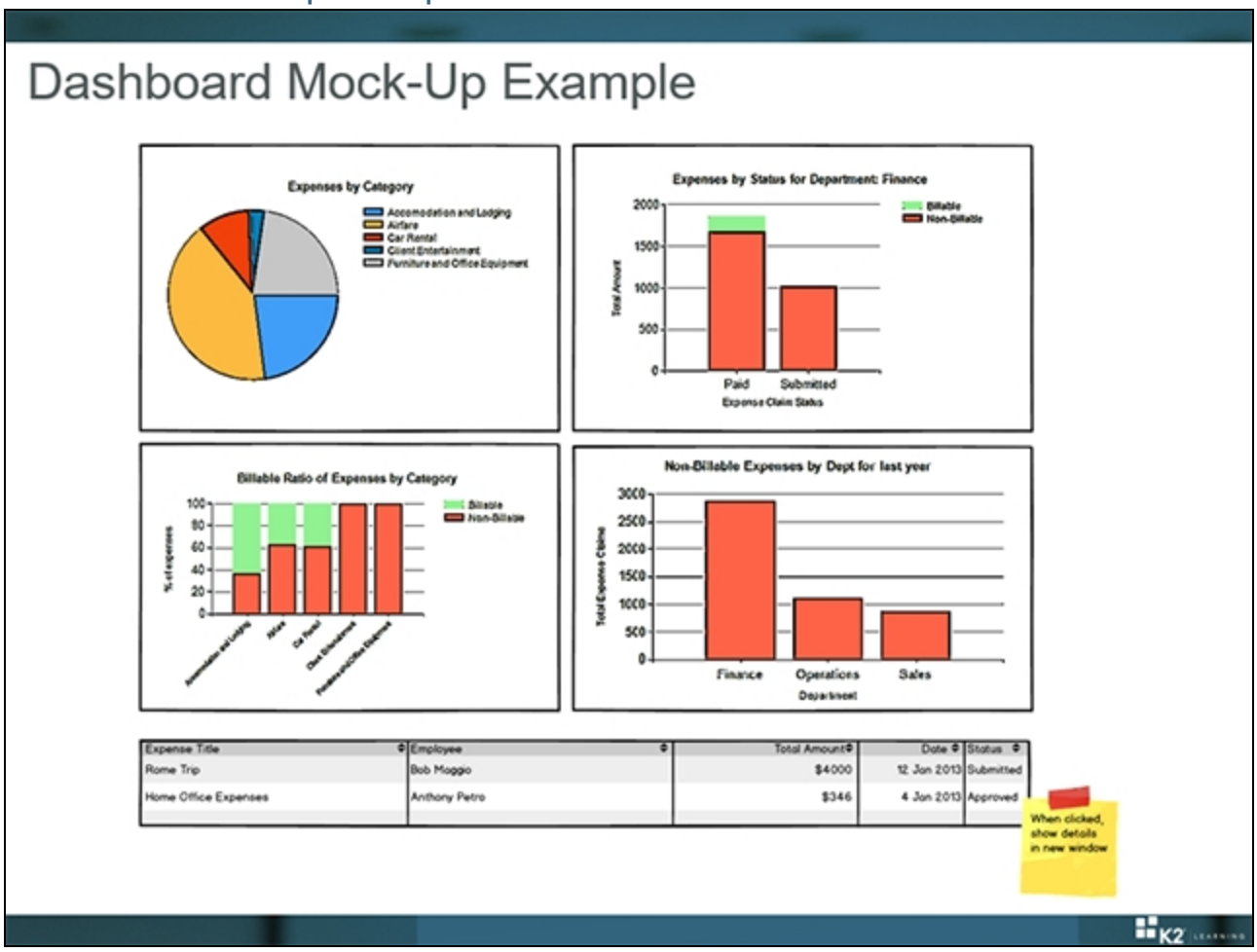
Finally, in the box marked, "**4. Report Rules**" determine:

- Parameters - last year, month, quarter
- Grouping/Summary - by department or business unit
- Sorting - department name
- Filtering data - do not show rejected expenses

Use the KISMIM (keep it simple, make it meaningful) rule as you build out each scenario.

As we did before, this shows an example of building Report models. It is similar to the Forms model except that the focus is slightly less on behavior and more on the information that the Report is intended to provide.

Dashboard Mock-Up Example



This is an example of a mock-up for the dashboard component of our sample expense claim approval scenario. Before you build your dashboard diagram, remember to identify:

- Who will "consume" or use the report
- What is the most important information
- Why and how often will the audience review your report

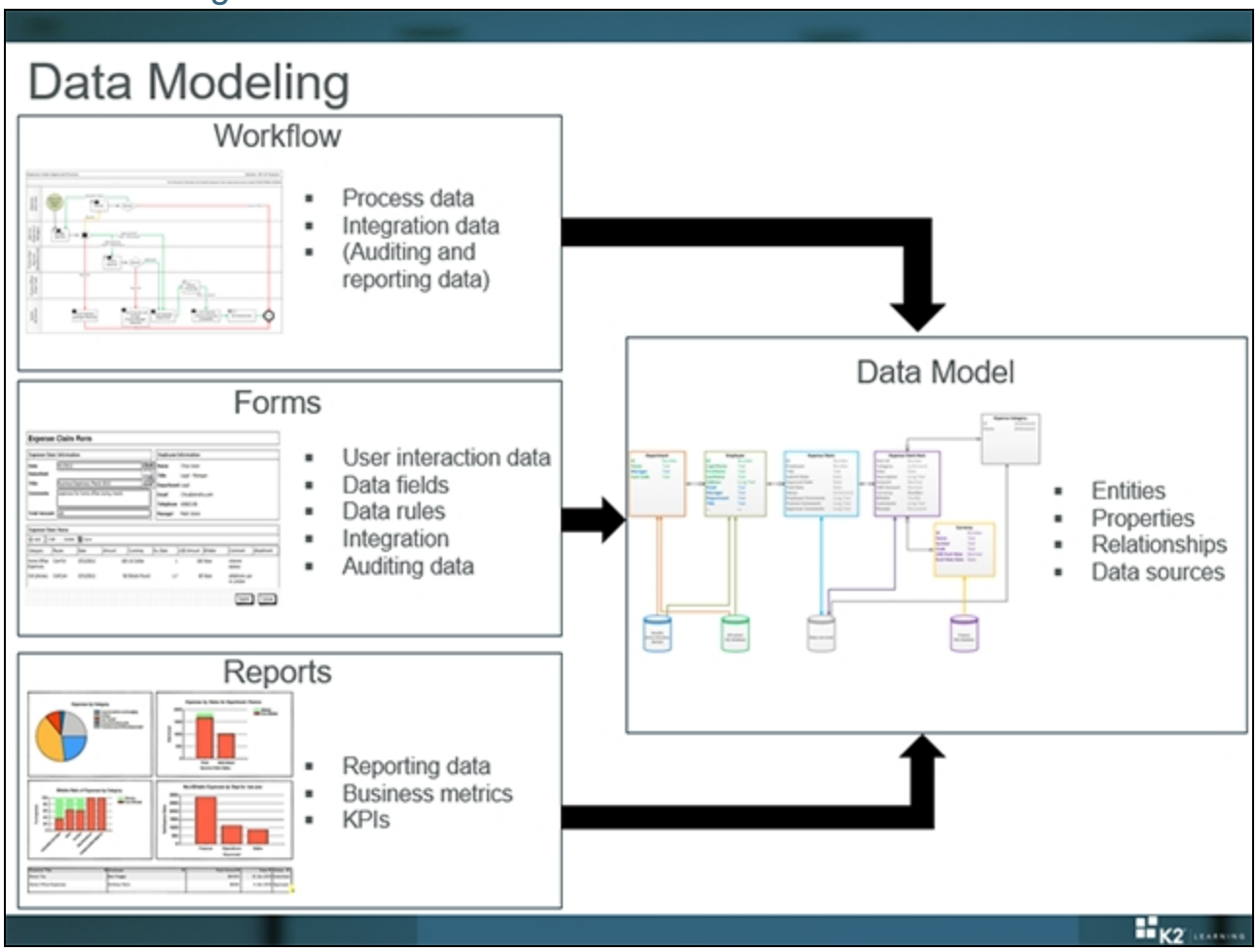
When you build your mockup, consider:

- Titling - for the form as a whole, typeface, size, style, justification, and color
- Labeling - of charts, graphs, and tables
- Layout - the arrangement of report elements
  - Diagram position on report - also consider white space between diagrams
  - Size of each diagram - relative to other elements
- Number of diagrams in the report - how many should be in the report
- Data used in each diagram - what data and how much should be included
- Types of diagram
  - Pie chart
  - Bar graph
  - Table
- Table headings and formatting - typeface, size, style, justification, and color
- Number of displayed lines in each table
- Data points in the table
- Amount of data per diagram
- Diagram legend
- Use of color

- Other graphical considerations
  - Borders
  - Border line weight
  - Border color

Before, during, and after you complete your mock-up, take time to check its design against your model. Don't forget to involve visually literate people (e.g. graphic designers) in the process.

Data Modeling



When it comes to data modeling, we should do the Workflow, Forms, and Reports first so that we know what data is important to the solution. The diagram shown here gives examples of what modeling tells you at this point about your data needs. This information, like tributaries feed into your data model with its entities, properties, relationships, and data sources. Let's look at each category:

Workflow

- Process data
- Integration data
- Auditing and reporting data

Forms

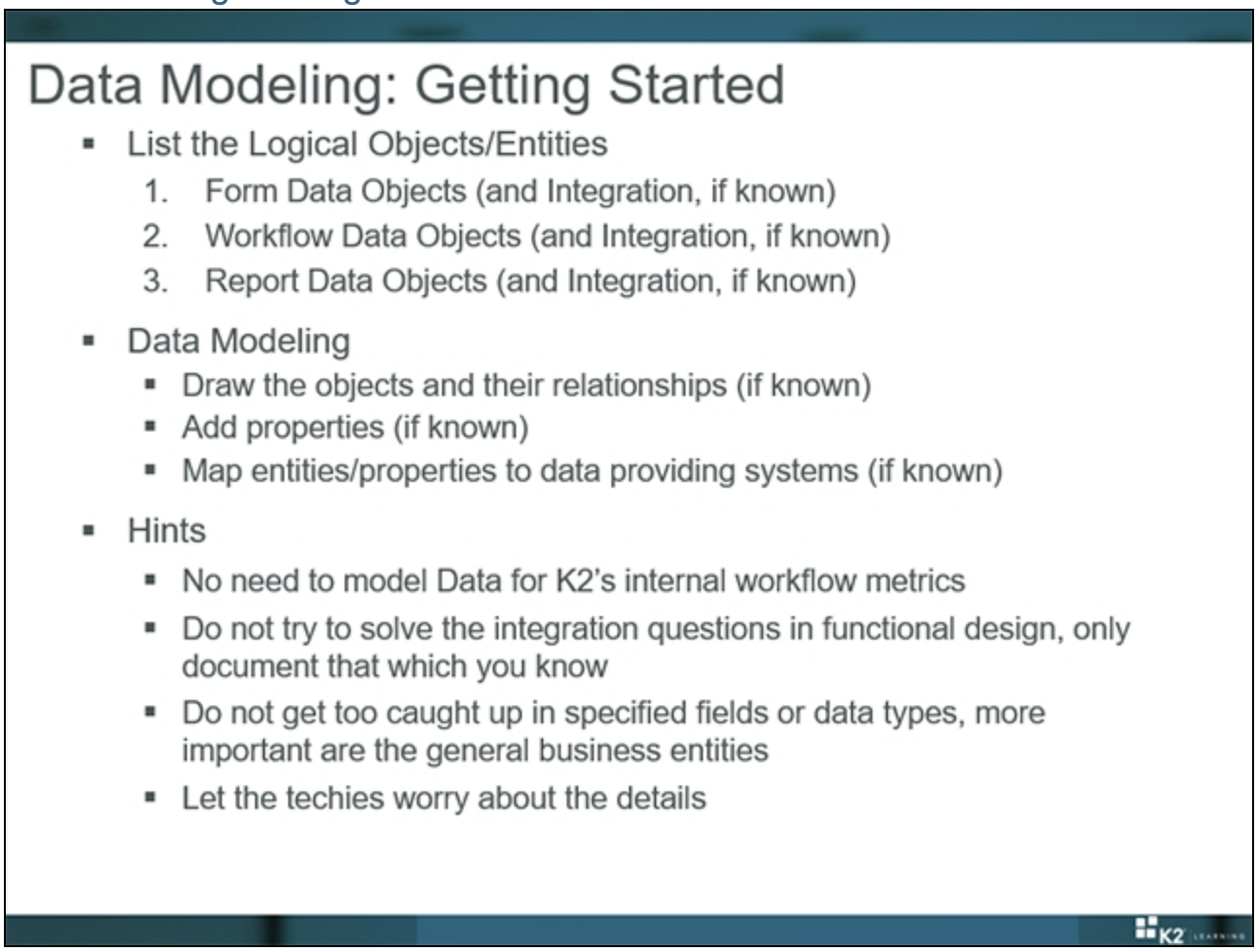
- User interaction data
- Data fields
- Data rules
- Integration
- Auditing data

Reports

- Reporting data
- Business metrics
- KPIs

Using technologies like SmartObjects means it is possible to re-use the same data entities between components, but first you need to know what data is required.

Data Modeling: Getting Started



Getting started with data modeling involves listing the logical entities or objects. Think of these as the building blocks for constructing the Workflows, Forms, and Reports you are developing. In our expense claim scenario we used SmartObjects as a data access component, but the underlying data was mostly stored in an SQL database, with some properties coming from SharePoint and some from ActiveDirectory.

Some examples of entities include:

- Department
- Employee
- Expense claim

Moving ahead with data modeling then involves:

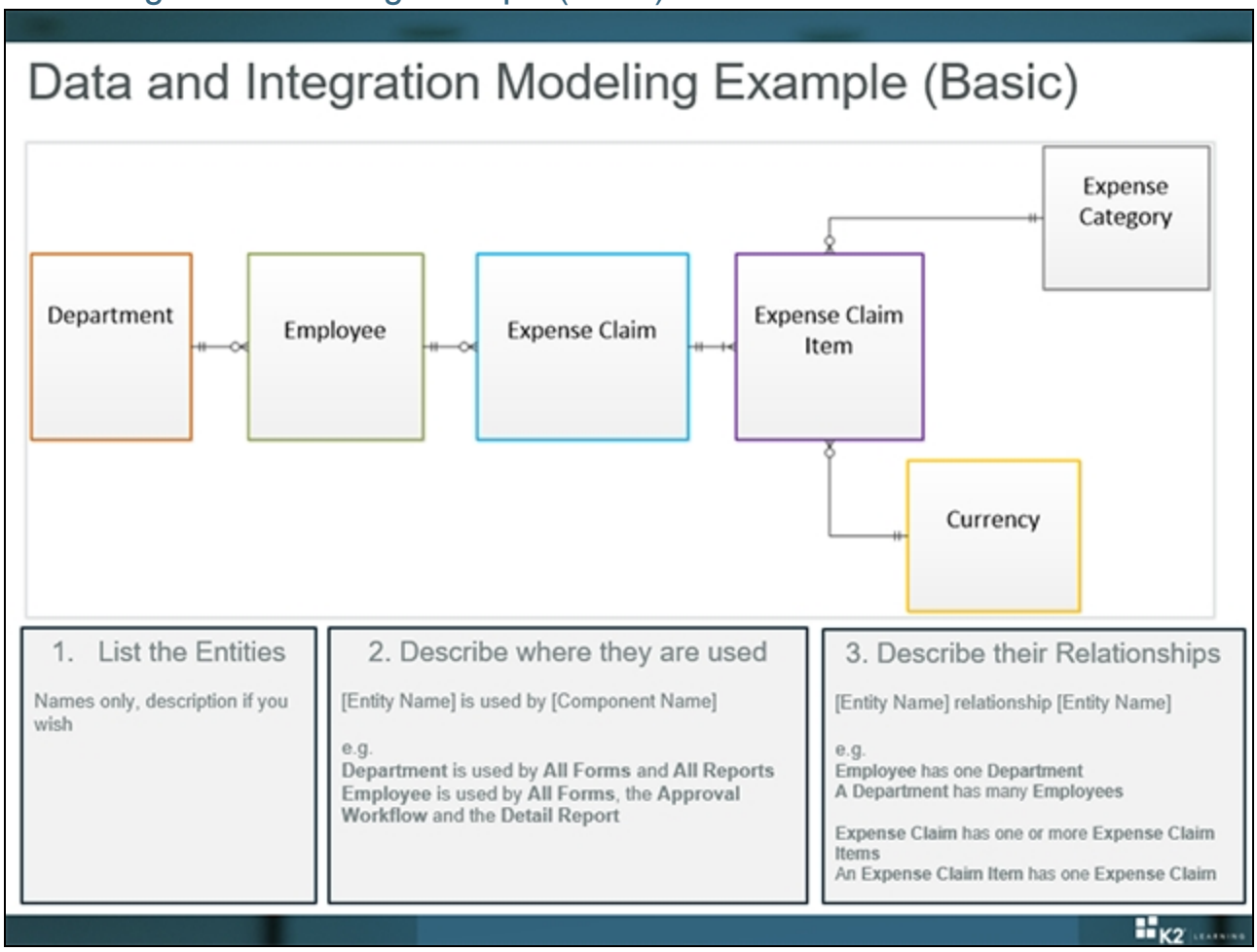
- Drawing the objects and their relationships
- Adding properties
- Mapping the entities and properties to data providing systems

Think of these steps as defining the "where" and "association" of data modeling. Consider these tips when you develop your data model:

- Don't bother to model Data for K2's internal workflow metrics
- Don't try to solve the integrations questions in functional design. Just document what you know at this stage.
- Don't get caught up in specified fields or data types. The business entities are more important.
- Let the "techies" worry about the details.

On this final point, leave the details about database queries, servers, and data structures to the IT professionals.

Data Integration Modeling Example (Basic)



In this basic stage of Data modeling and integration, you begin by listing the entities. That is, what are the names of the business entities involved? You may choose to add a description as well, but at this point it is a good idea to at least identify the entities.

Next you describe where each entity is used. In this example we state that:

- **Department** is used by All Forms and All Reports
- **Employee** is used by All Forms, the Approval Workflow, and the Detail Report

In the following step, describe their relationships. Continuing with this example we identify:

- **Employee** has one Department
- A **Department** has many Employees

At this point you do not have to specify where data comes from, but a model like this can help designers start to understand what data objects are required.

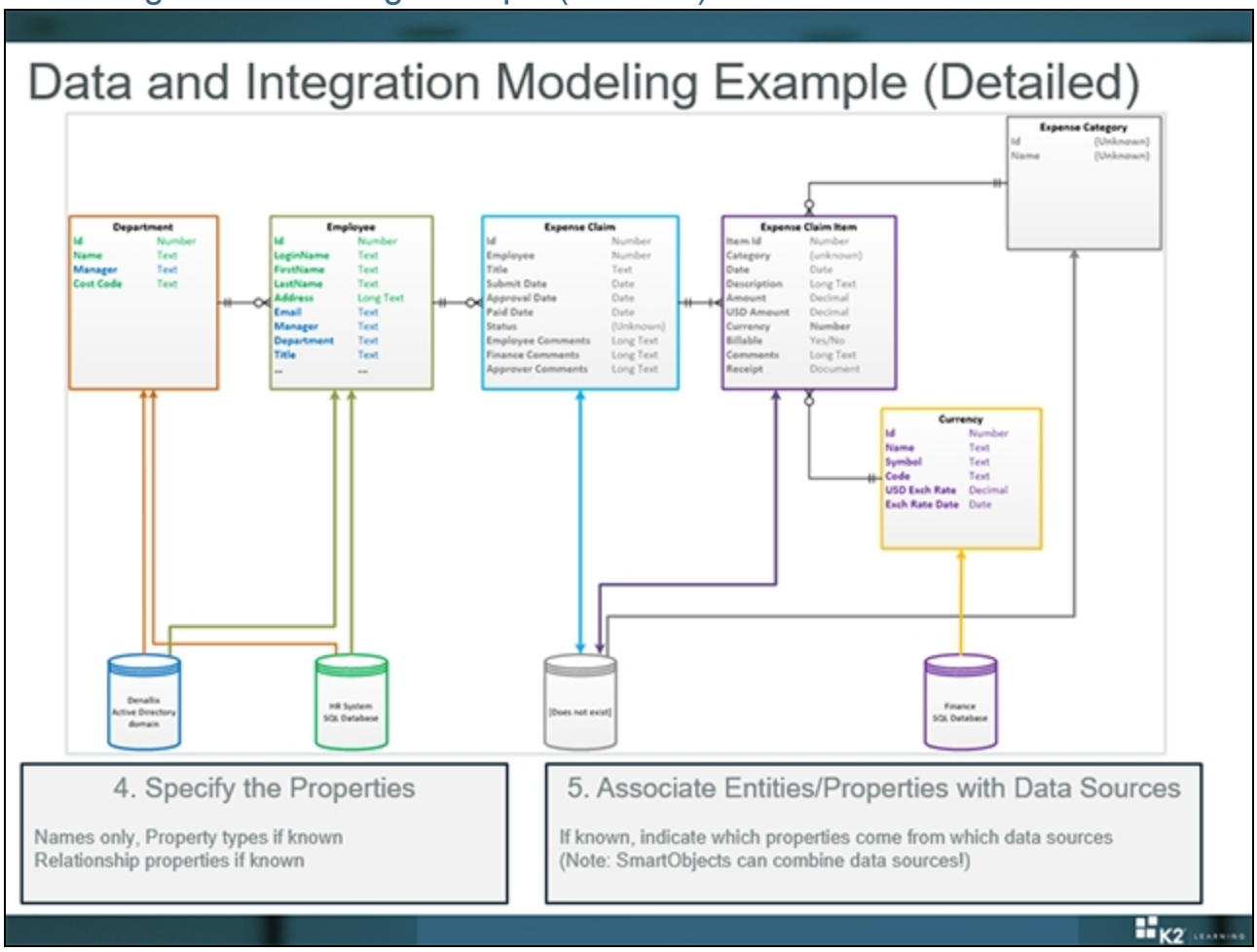
You can also specify the relationships between data if known. This will help designers know how to implement the data and how the data should be rendered on Forms and Reports.

This is a basic level of modeling because:

- It doesn't require IT involvement or technical detail
- You are dealing in abstract entities and objects
- You are moving from the general to the specific

Notice the difference between Basic modeling and the level of detail in the following Detailed level.

Data Integration Modeling Example (Detailed)



In step 4 of our Data and Integration Modeling process, we move beyond the basics into the details of the properties for business objects and the data sources that provide those properties if known. Step 4 is where you specify properties, including:

- Names
- Properties, if known
- Relationship properties, if known

In the example above, refer to the Department entity. The left column contains these **properties**:

- Id
- Name
- Manager
- Cost Code

The corresponding **property types** are displayed in the right column:

- Number
- Text
- Text
- Text

Continue adding properties and types for each entity. When you have completed step 4, continue with step 5: Associate Entities/Properties with Data Sources. Step 5 in particular requires IT expertise in data sources. Here you will identify which properties come from which data source. As you complete your model, it may begin to look a bit like our example above.

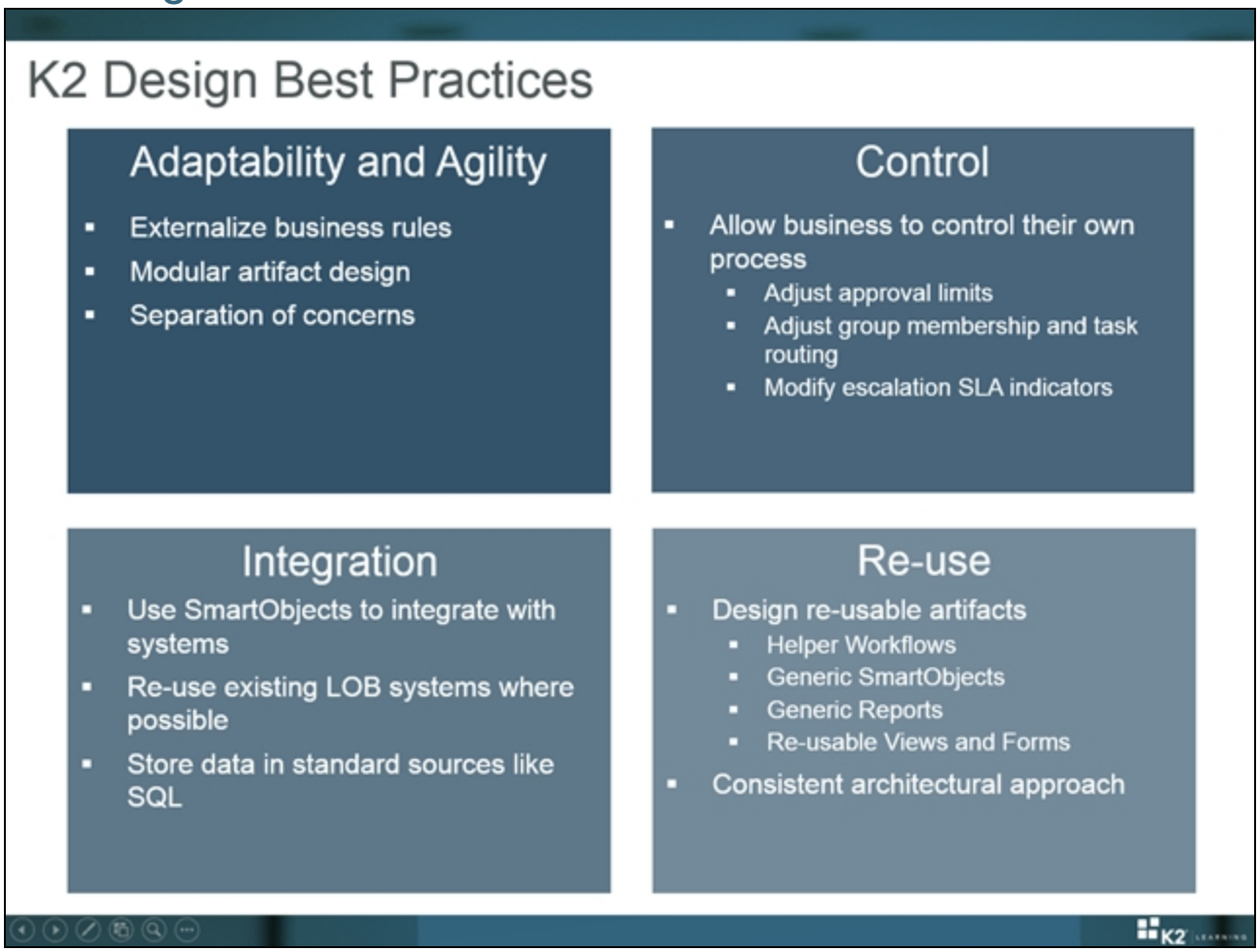
WORKSHOP SESSION 3: MODELING FORMS REPORTS DATA

Note
The workshop exercises in this learning module use several resources, for example template documents and workshop guides. If you have not done so already, please download the following file from K2's help site: http://help.k2.com/files/10701.
Once the zip file is downloaded, please extract the contents of the zip file to your computer. (Alternatively, you can also extract the zip file to the virtual K2 environment provisioned for you for this training course.)

In this workshop we will look at some principles of modeling Forms, Reports, and Data for K2 applications. Similarly to the process modeling we did in the previous exercise, it is recommended to follow a standard procedure to model the other components (Forms, Reports and Data) of K2 applications. You can follow the same Descriptive > Analytical > Specification strategy to gather requirements for these components. Your organization may already have patterns and practices in place for modeling these components and that is why we separated out the workflow modeling exercise from the Forms-Reports-Data modeling exercise.

Note
After downloading and extracting the resources for this module as described in the note above, you can find the sample documents for this workshop at the following location:
[Extracted Location]\Workshop Resources\Workshop Session 3 - Modelling Forms Reports Data

K2 Design Best Practices



Finally, let's wrap up with some best practices and guidelines for design and implementation of K2 applications.

Adaptability and Agility

Where possible, it's a good idea to allow business to control the rules and parameters in the solution. This is most often achieved by externalizing the business rules in the solution and then giving the business some interface where they can manage their rules. Here is a typical example. Suppose there is an SLA that determines the number of hours that should elapse before a particular workflow step sends a reminder email. That SLA may change over time, so perhaps it would be a better idea to expose the number of hours in a SharePoint list so that an authorized user can change the number of hours outside of the workflow definition. The workflow developer would implement a K2 SmartObject that exposes the SharePoint list, and then use the SmartObject in their escalation rule to return the number of hours. That way, business can maintain the business rule themselves and the workflow developer does not need to make any changes to the workflow definition.

When generating content (for example, email bodies), it is a good idea to generate that content dynamically, perhaps by using template content stored in a file or SmartObject. This way, the email body can be changed without having to modify the workflow definition.

Finally, it is a good idea to pull data from external systems where possible and appropriate, to reduce the possibility of data duplication. It is very easy to create SmartObjects to interact with external systems. The developer need only store a record ID in their workflows to be able to retrieve the data from the underlying system in real time when it is needed by the workflow.

Control

It is a good idea to externalize the business rules in a workflow or process so that they can maintained outside of the actual process definitions. This will make the solution more controllable without requiring any redevelopment work.

Reports should be easy to use and relevant to the business case, so that the information they provide is usable. There's no sense developing the ultimate report with wonderful graphics, if it doesn't help the business answer its questions.

Exception cases and escalations should be handled where appropriate. It is easy for a business to get carried away with all the possible exception cases, or to go overboard with complex or intricate escalation rules just because they can. Be pragmatic about catering for these requirements. Remember Pareto's 80/20 principle.

Integration

If the data and systems that need to be integrated into the solution already exist, try to integrate with these systems as much as possible rather than rebuilding the same functionality.

Although this may be tricky, try to design the system with future integration in mind. Using SmartObjects or APIs is a good way to abstract the implementation of a data store from the usage of that data store. We recommend using SmartObjects wherever possible to build this integration, since they are such a central part of K2 and can be easily consumed in K2 components, ASP.NET, and SSRS reports.

Re-Use

Try to specify components in the solution for re-use so that you can leverage existing functionality or re-use the artifacts in this solution in other projects. For example, if developers will be building APIs to integrate with K2, it is good idea to make these APIs generic so that they can be used in other projects. SmartObjects are another component that other K2 projects can easily re-use.

Where possible, reports should be generic so that other projects can re-use them. Of course, this is not always possible: perhaps a good approach would be to provide a set of generic reports as part of every K2 project and then implement specific reports for the requirements not met with the generic reports.

When using K2 smartforms, it is a good idea to separate the user interfaces into generic, re-usable parts (called "Views" in K2). Here is an example: suppose that there is a requirement to display employee data on a Form in the current requirement. This type of data is often required in other projects, so it would be a good idea to create an "Employee Details" view for this data so that the view can be re-used in other projects. You don't need to implement the view as part of the specification phase. Just design the user interface so that the employee details can be separated into a separate section of the form.

Finally, it is a good idea to implement best practices, development standards and naming conventions in K2 projects. The point is to make sure that K2 projects in your organization are consistently implemented.

Review and Q&A

We have covered quite a lot of information. Take a moment to review the main topics we spoke about in this session. If time permits, consider your own requirements and how these topics would apply to them.

Students should now know:

- How do you qualify a requirement for K2
- What types of questions you should ask in the Initial Discovery phase
- How to develop functional requirements for K2 projects
- How to gather functional specifications for Forms, Workflow, Reports, and Data
- What are the questions you should ask?
- How to use the Descriptive -> Analytical -> Specification technique
- How to discover and model Forms, Reports, and Data
- K2 design best practices

100.CWL: Reporting in K2

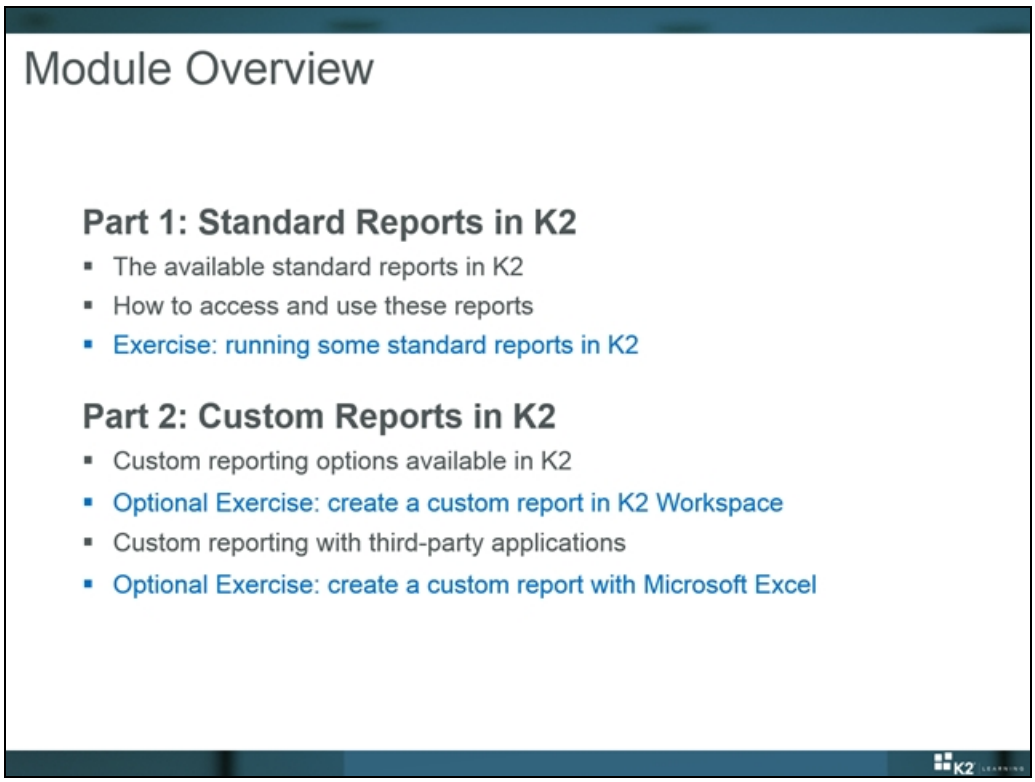
Video

The *100.CWL: Reporting in K2* training module explains how to use the available standard and custom reporting in K2 to report on workflows. The module should take around 1 to 1.5 hours to complete, depending on how many of the hands-on exercises you choose to do.

This module covers the following concepts:

- The standard reports that are available in K2 and where you find and run those reports
- Using the K2 View Flow report
- Creating custom reports in K2 Workspace and K2 smartforms
- Creating custom reports with other third-party reporting tools

Module Overview

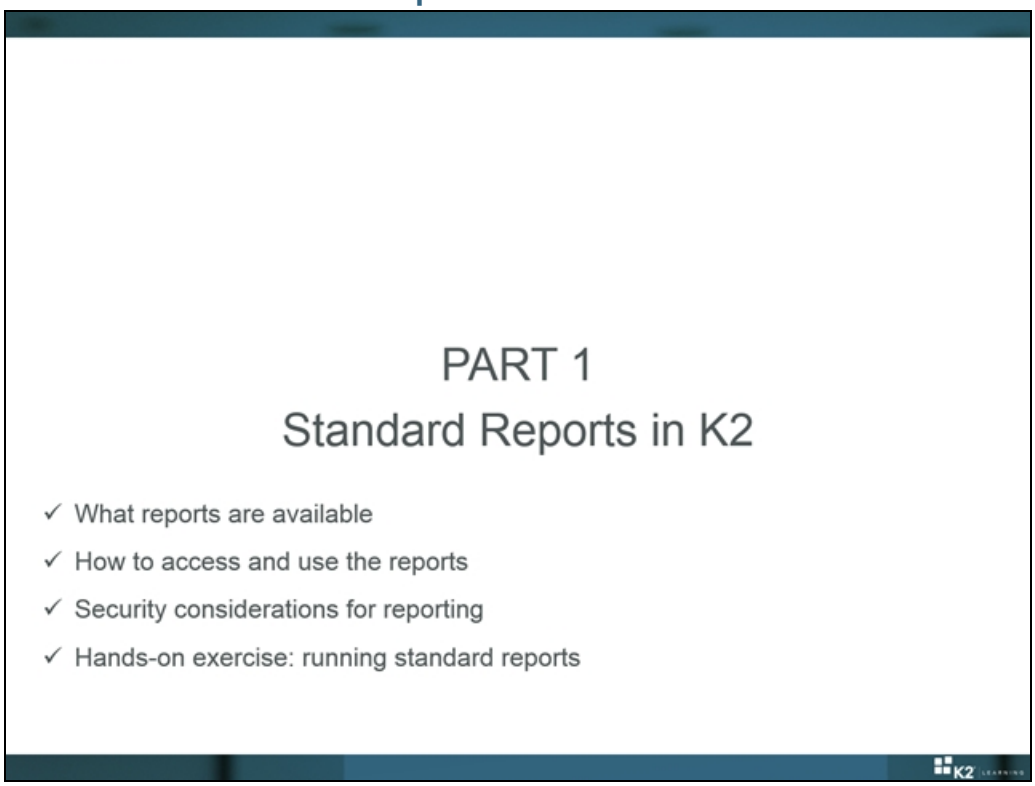


Video

This module consists of two parts: Part 1 (standard reports in K2) deals with the available standard reports that are used to report on K2 workflows. Part 1 has some exercises where you will learn how to work with the standard reports in K2.

Part 2 deals with creating custom reports in K2, using either built-in K2 custom reporting tools or third-party reporting tools. This part is optional, and each section ends with an optional exercise. (The exercises are optional because not all organizations may wish to create custom reports).

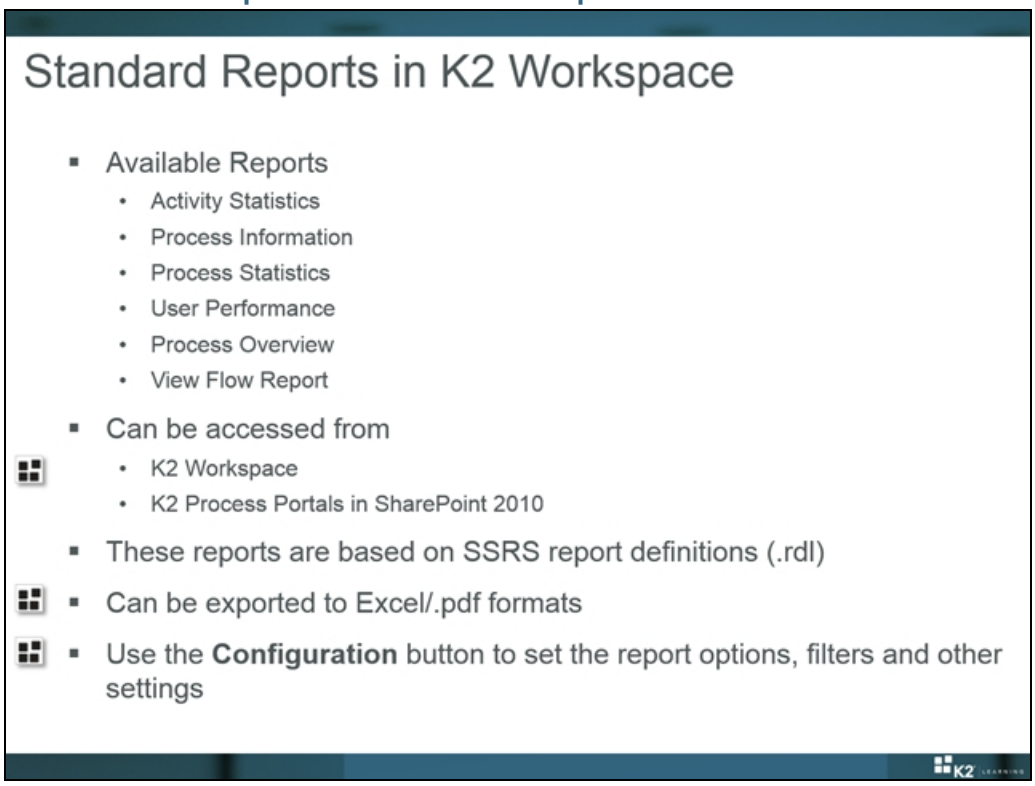
Part 1: Standard Reports in K2



Video

In Part 1 we will look at the standard reports that are available in K2, how to access and use these reports and some security considerations that affect how reporting data is shown to users. At the end of Part 1 there will be a hands-on exercise followed by a mastery check.

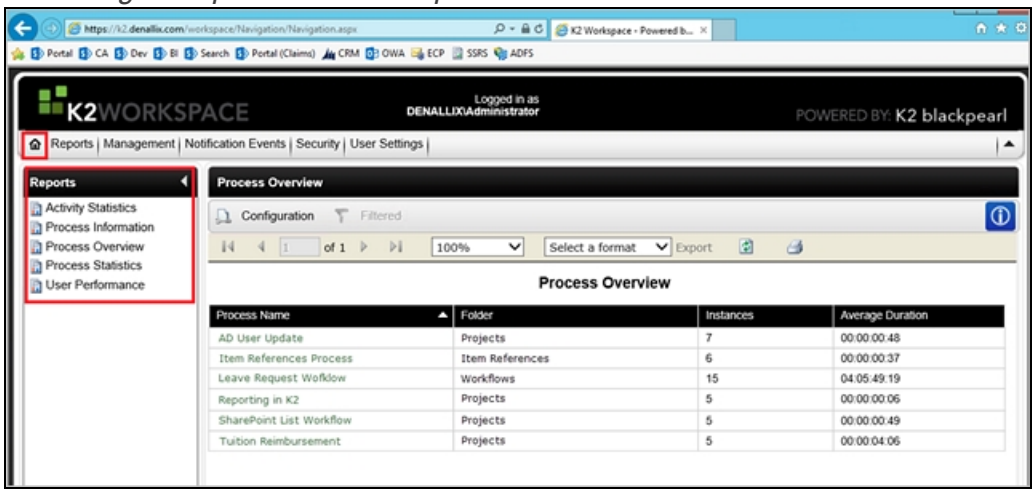
Standard Reports in K2 Workspace



Video

The K2 Workspace is a web-based application that is commonly used to report on and administer a K2 environment. The application is opened by browsing to a particular URL (ask your K2 administrator for the location of the K2 Workspace in your environment). By default, the workspace is always opened on the landing page, which contains the K2 worklist and a series of links on the left-hand side for running the standard K2 reports.

Accessing the reports in K2 Workspace

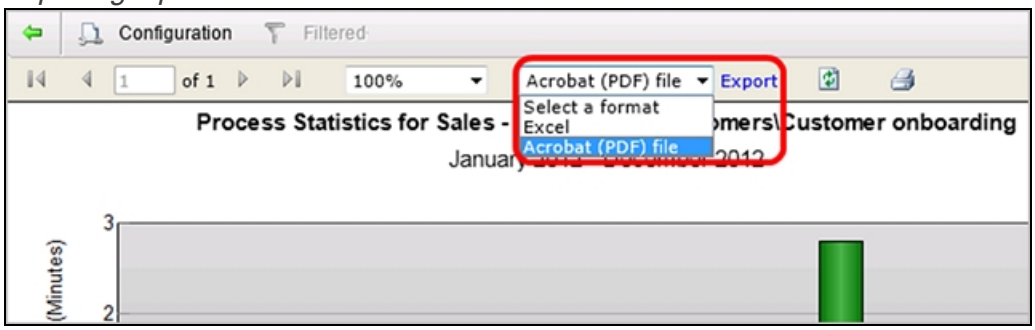


K2 comes with a selection of standard reports that can be used to report on workflow history. These reports can be used for Business Intelligence (BI) and for Business Activity Monitoring (BAM). BI generally refers to reports used for statistical analysis and historical reporting. Examples include running a report to identify the bottlenecks that occur in a particular process so that corrective action can be taken, or determining what the peak usage intervals for particular workflows are so that appropriate staffing arrangements can be made to handle the additional load. Standard Reports include high-level overviews of usage statistics (for example, average durations or number of instances of a Process) or very detailed reports that provide the ability to "drill-down" into a Process Instance for Activity Instance details and audit trails.

BAM generally refers to run-time reporting like process status reports and real-time "dashboards". An example of BAM is to run a visual report to understand where a particular instance of a process is right now and which users have made which decisions that caused the process to follow a particular route.

The reports are based on the industry-standard SQL Server Reporting Services (SSRS) .rdl format. The report output can also be exported to another format like PDF or Excel.

Exporting report data to another format

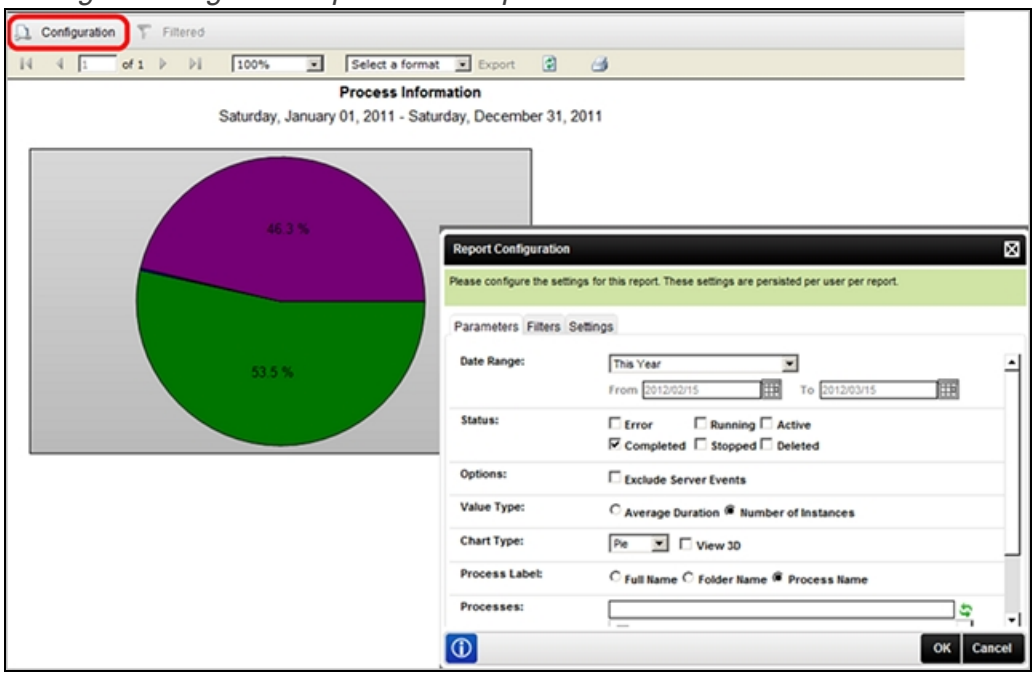


Available Reports

K2 has several Standard Reports that provide a well-rounded overview of what is occurring with your workflows. Standard Reports can be used for Business Intelligence (BI) and would include information such as statistical analysis (number of Processes started, average duration of an Activity) or historical reporting (number of Processes started in a given time frame, peak usage). Standard Reports can also be used for Business Activity Monitoring (BAM). For example, running a report to see where the Process is right now, or which users actioned a user task and the path that the Process took based on that action.

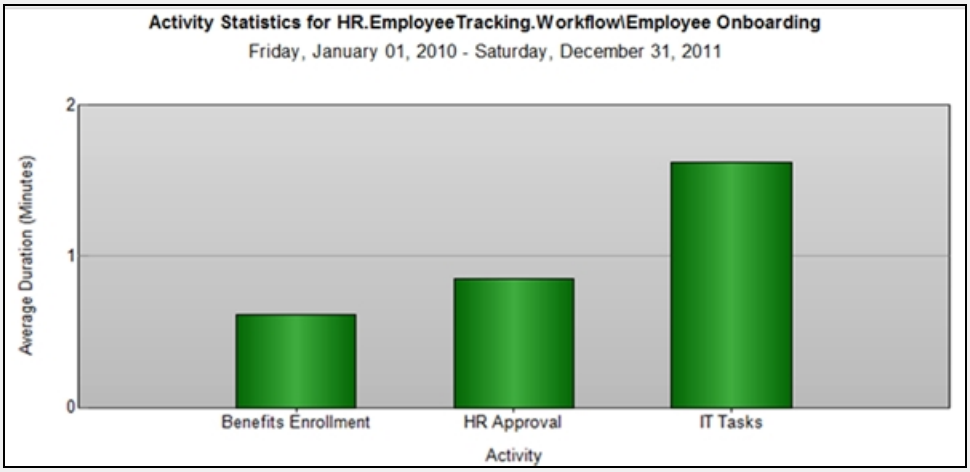
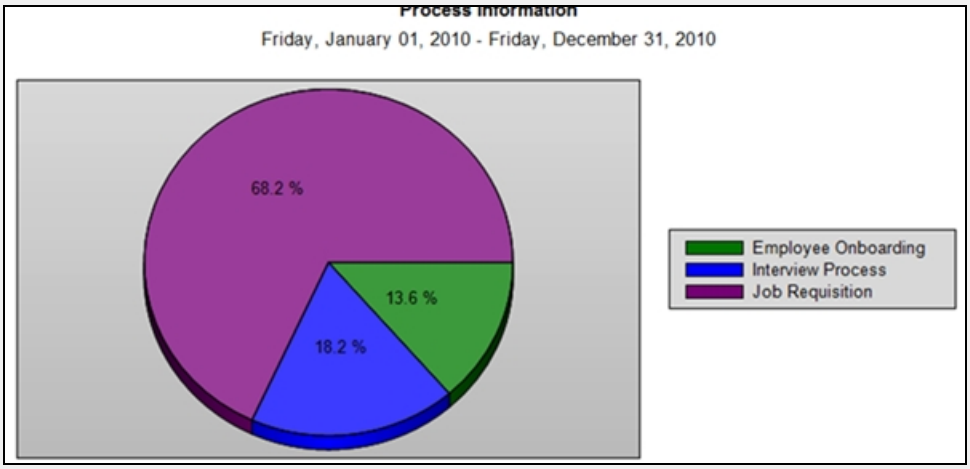
Standard Reports differ from Custom Reports in that they typically have little flexibility in their display or the data contained in the report. While there are settings you can configure to add parameters and filters, it is not possible to add custom fields or create custom layouts in Standard Reports.

Setting the configuration options for a report



The following table provides an overview of the five Standard Reports found in K2 Workspace.

Report	Use Case
Activity Statistics	This report is typically used to discover bottlenecks in a specific process, or to identify which tasks have the most load in a workflow. In the image below, we are looking at the average time it takes for each Activity (or step) within a Employee Onboarding Process. Notice how the IT Tasks Activity con-

	sistently takes much longer to complete over the other two tasks. 
Process Information	This report compares different Processes over the same time frame, and is most often used to compare the peak use of different Processes or the average time taken to complete different Processes. In the image below, we can easily see that the Job Requisition Process is taking much longer on average to be completed. This report would be helpful to determine if a Process needs to be reworked or otherwise needs some intervention. It might also lead you to looking at the Activity Statistics for the long-running Process to determine if any Activities need intervention. 
Process Overview	The Process Overview Report lists all instances of a workflow (for active workflows as well as completed workflows) and allows you to drill down into the Process Instance data and statistics. This report is very useful for development, testing, troubleshooting and investigation of Process Instances, and for performing audit trail investigations of Processes. This report illustrates why the Folio is important: it allows us to differentiate each Process Instance with some human-readable unique identifier. Notice too, that we can see the Status for each Process Instance, some being completed and some in "Error" status. If you click on the Folio column, you can drill down into that instance to see a listing of the activities, events and audit trails for that workflow instance.

Folio is often used to distinguish process instances from one another

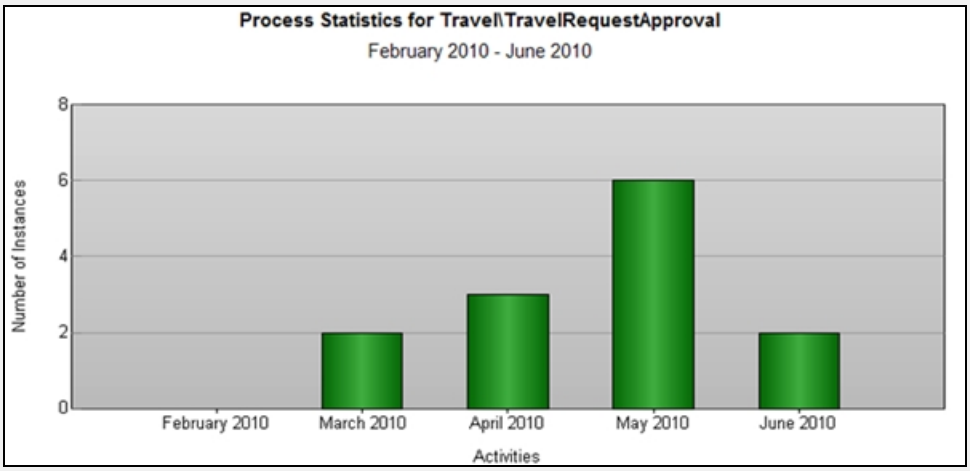
Process Instances

Process: HR.EmployeeTracking.Workflow\Job Requisition

Process Folio	Originator	Status	Priority	Start Date	Finish Date	Duration
[262] Job Req - Network Administrator	K2:DENALLIX\ADMINISTRATOR	Error	Medium	10/12/2011 8:10:33 AM		00:01:23:41
[261] Job Req - Admin Assistant	K2:DENALLIX\ADMINISTRATOR	Error	Medium	10/12/2011 8:08:47 AM		00:01:25:28
[225] Job Req - Customer Account Manager	K2:DENALLIX\JAMES	Completed	Medium	1/24/2011 8:32:58 AM	1/24/2011 8:34:28 AM	00:00:01:30
[222] Job Req - Financial Analyst	K2:DENALLIX\ANTHONY	Completed	Medium	1/24/2011 8:16:47 AM	1/24/2011 8:19:24 AM	00:00:02:36
[219] Job Req - Assistant Corporate Attorney	K2:DENALLIX\CHRIS	Completed	Medium	1/21/2011 7:43:13 AM	1/21/2011 7:47:05 AM	00:00:03:51
[203] Job Req - QA Engineer	K2:DENALLIX\ADMINISTRATOR	Completed	Medium	1/14/2011 12:52:10 PM	1/19/2011 6:01:24 PM	05:05:09:13
[202] Job Req - Quality Assurance Manager	K2:DENALLIX\ADMINISTRATOR	Completed	Medium	1/14/2011 12:50:39 PM	1/19/2011 6:05:04 PM	05:05:14:24

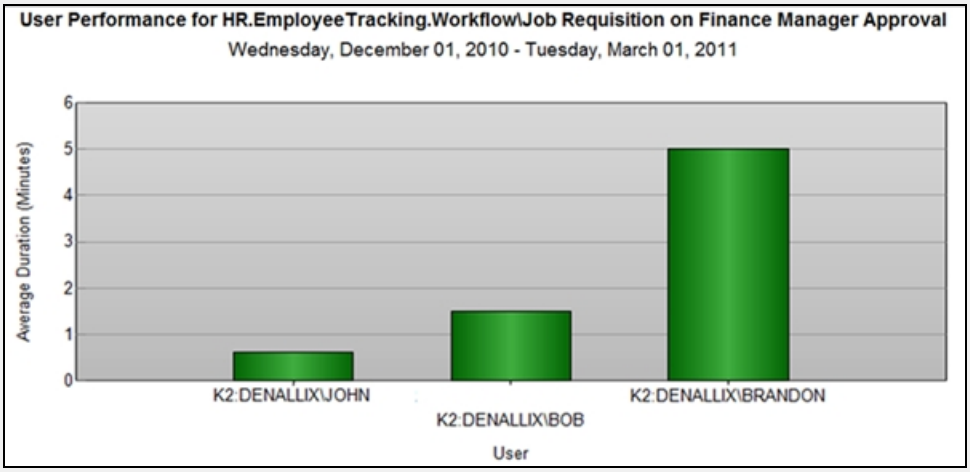
Process Statistics

This report displays a count or duration graph of a specific Process grouped into selected intervals. This report can be used to determine peak-usage times for the Process. In the image below, we have configured the report to show the average duration of the Travel Request Workflow Approval process over several months. As more and more workflows are processed, this information would be valuable in determining peak usage times.

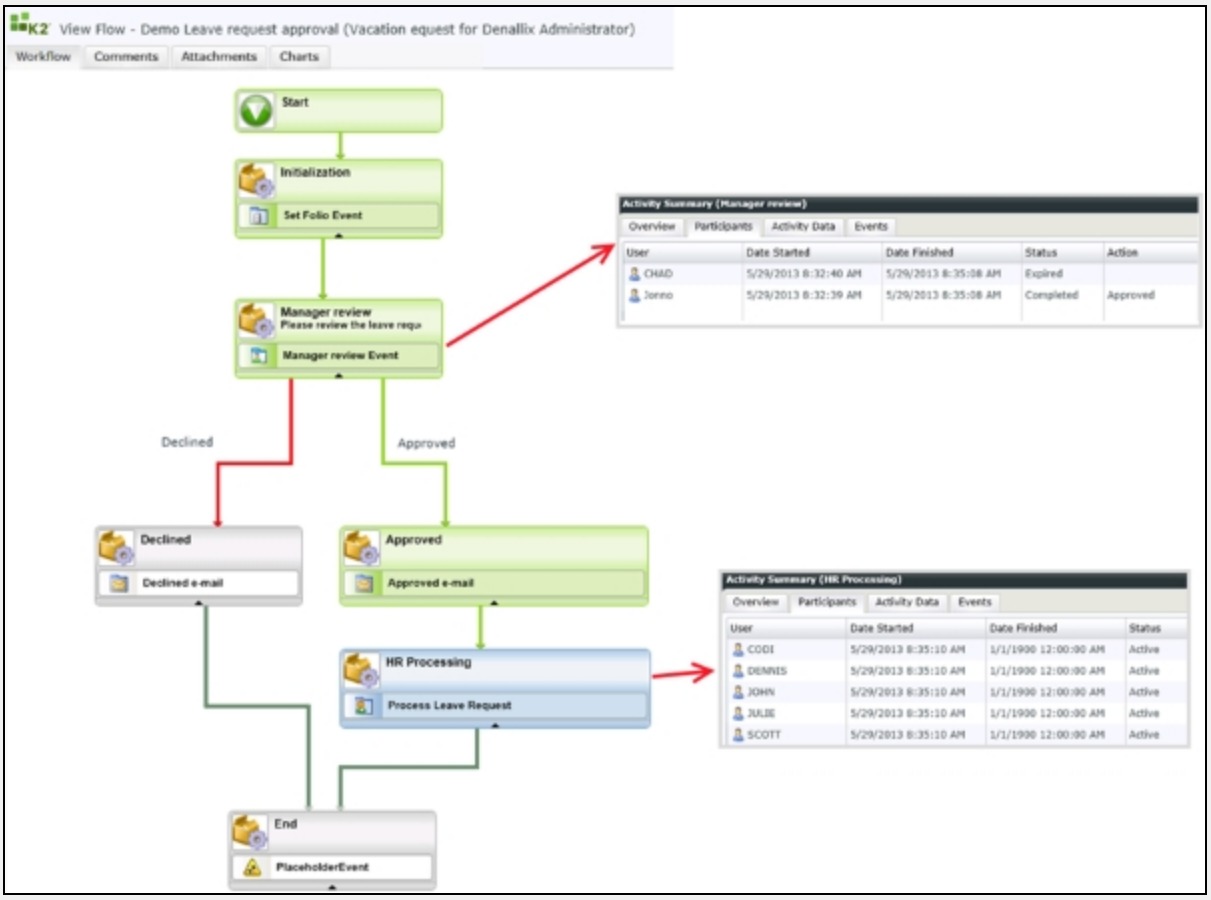


User Performance

The User Performance Report is an average duration or number of tasks competed for all users that performed a specific user task in a Process. This is useful in team-management scenarios (for example, identifying which users in a team do the most work) or in user performance management (for example, understanding how long a particular user takes on average to complete a specific task). In the image below, we are looking at the number of Finance Manager Approval tasks each user completed for the Job Requisition process, over a 4-month timespan.



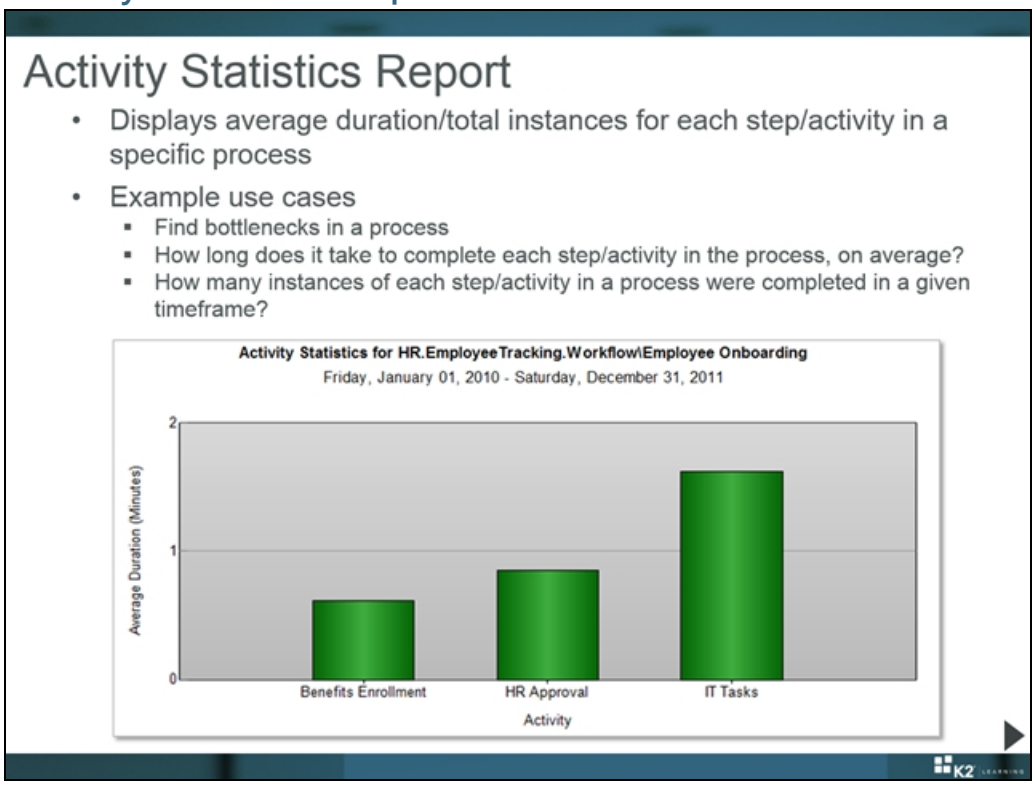
View Flow Report	This report displays the same content as the Process Overview Report (with a few minor differences), in a flowchart format and is a "live" view of a workflow instance: it will update automatically as the workflow progresses. In the second part of the Standard Reports tutorial, we will explore the View Flow Report and its features.



Summary

- Reports are available in K2 Workspace and in other interfaces like K2 for SharePoint 2010 Process Portals
- The standard K2 reports are workflow-centric and present workflow statistics and workflow tracking data
- These standard reports do not include business data
  - You can write custom reports with business data by using the reporting SmartObjects as we will see later
- Reports are based on SQL Server Reporting Services and can be exported to other formats
- Click the **Configuration** button to set reporting filters and other options

Activity Statistics Report



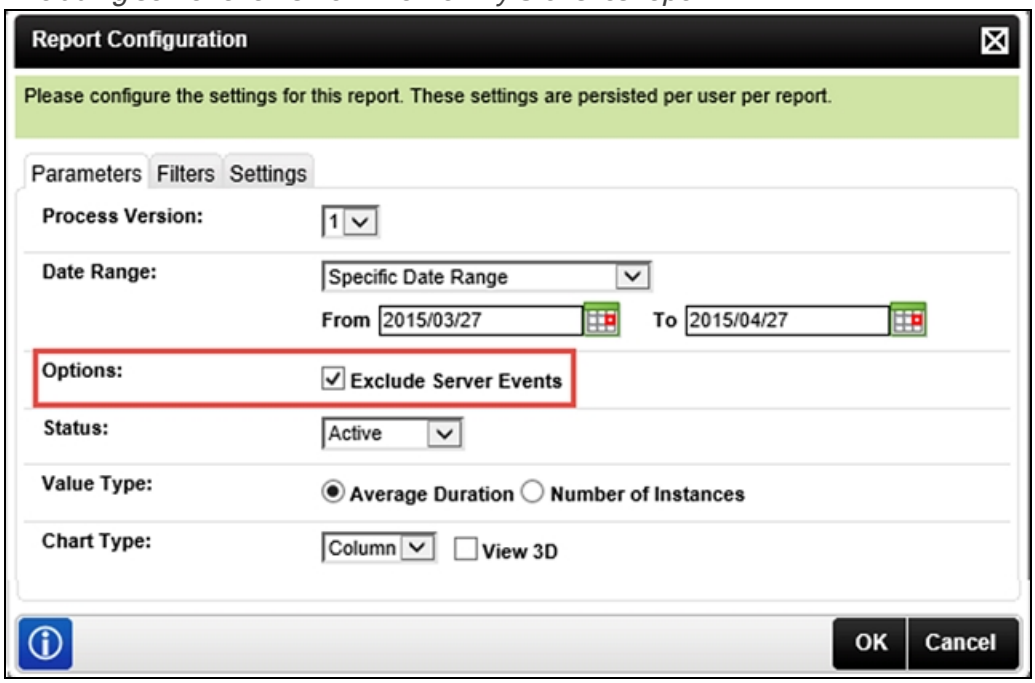
Video

The Activity Statistics report is used to query the “total number of instances” or ”average duration” for each step/activity in a workflow. It is most often used to determine which steps are the bottlenecks in a workflow (in other words, how long does it take, on average, to complete a step in the workflow?). You can also use the report to determine how many instances of each activity have existed over a given time span (how many instances of each step/activity in a process were completed in a given time frame?), which allows you to determine which activities are used most often.

The duration interval is measured from when the activity is started (assigned to a user for user tasks) until the activity completes. Note that it is not measured form the time a user actually opens the task.

You can use the report filter options to adjust the status of workflows included in the report (such as showing only Active workflows), and it is recommended to **exclude server events** from the report since these tend to complete very quickly and do not contribute useful data to the report.

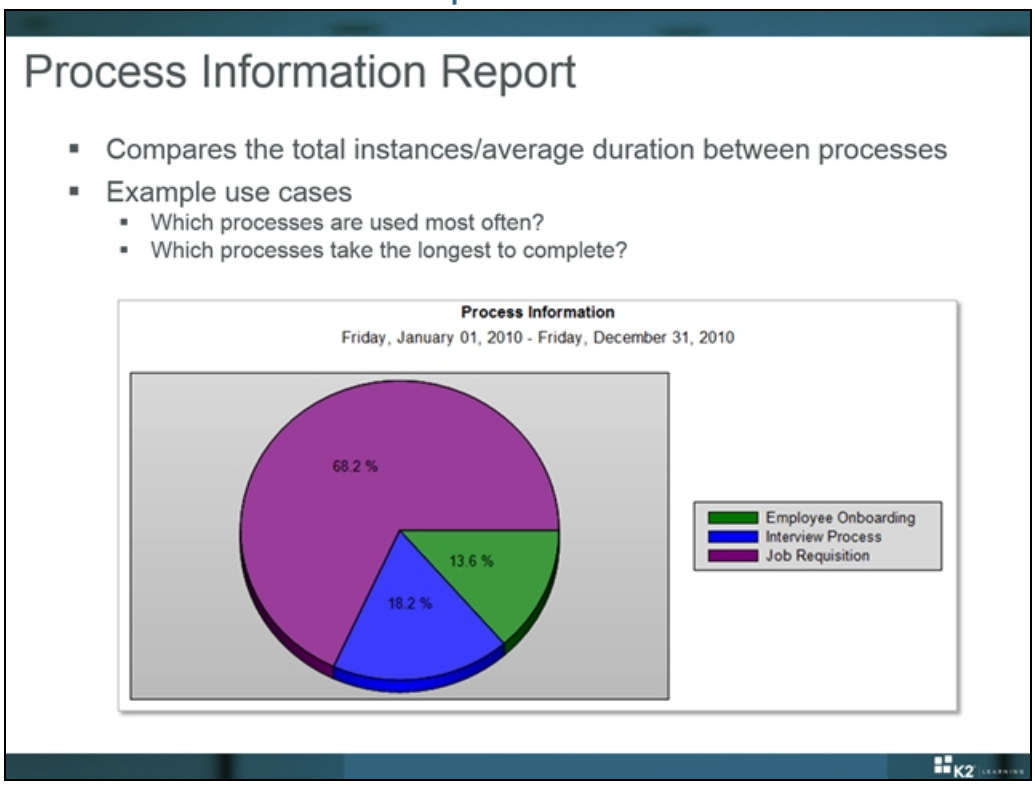
Excluding server events from the Activity Statistics report



Summary

- The Activity Statistics report is used to query the “total number of instances” or ”average duration” for each step/activity in a workflow
- Mostly used to determine which steps are the bottlenecks in a workflow
- Time is measured from when the activity is started (assigned to a user for user tasks) until the activity completes
- Can use the report filter options to adjust the status of workflows included in the report
- Recommended to **exclude server events** from the report

Process Information Report



Video

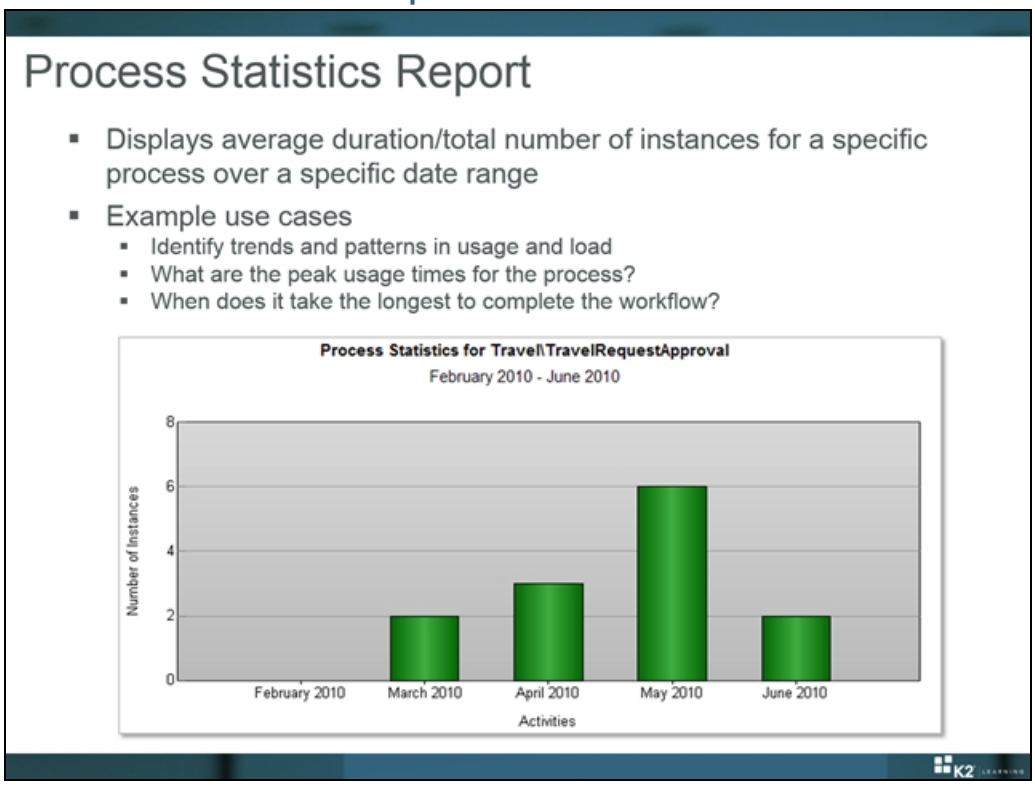
The Process Information report will display duration or volume comparison between different processes (e.g. which of our processes are used the most often? Which of our processes take the longest to complete?).

The duration interval is measured from when the workflow instance is started until the workflow instance completes. (A workflow completes when there is nowhere else for the workflow to go.) As with the Activity Statistics report, you can use the report filter options to adjust the status of workflows included in the report.

Summary

- The Process Information report is used to query the “total number of instances” or ”average duration” between processes
- Mostly used to compare how long different workflows take to complete or which ones are used most often
- Duration is measured from when the workflow instance is started until the workflow instance completes

Process Statistics Report



Video

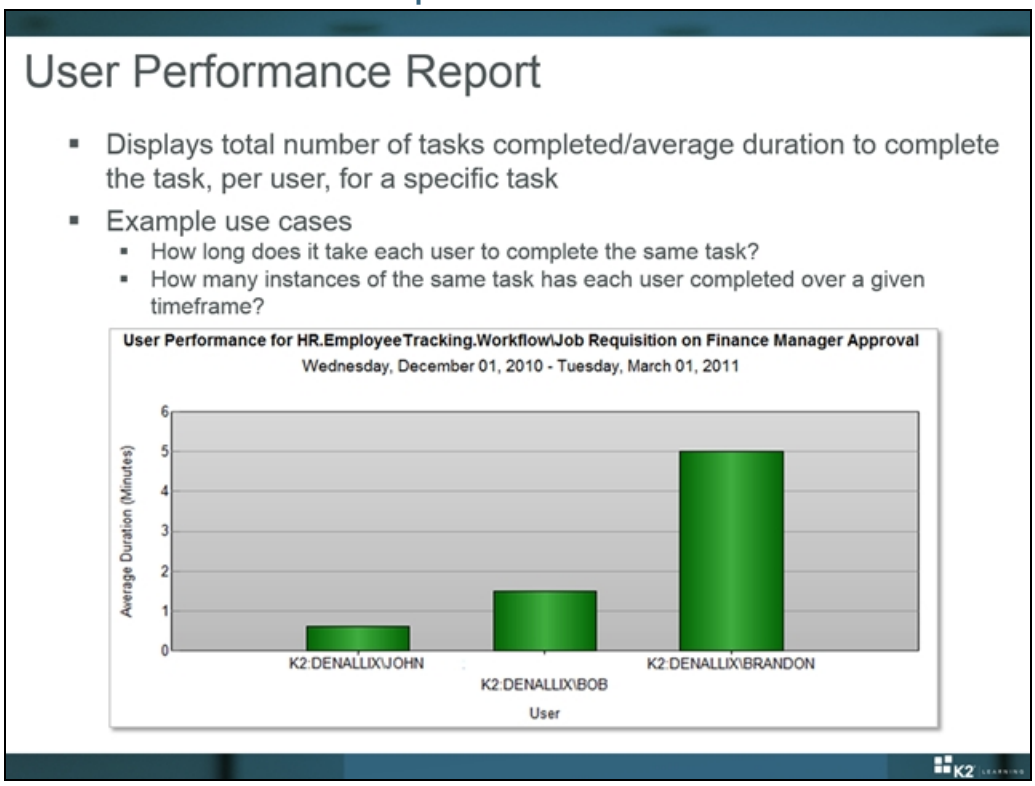
The Process Statistics report will display duration or volume information for a specific process over a given time frame. It is most often used to identify trends and patterns in usage and load for a particular workflow. Typical questions this report might answer include "What are the peak usage times for the process?" and "At what times does it take the longest to complete the workflow?".

The duration interval is measured from when the workflow instance is started until the workflow instance completes.

Summary

- The Process Statistics report is used to query the “average duration” or ”total number of instances” for a specific process over a specific date range
- Mostly used to identify patterns in the number of instances or the time taken to complete a workflow definition

User Performance Report



Video

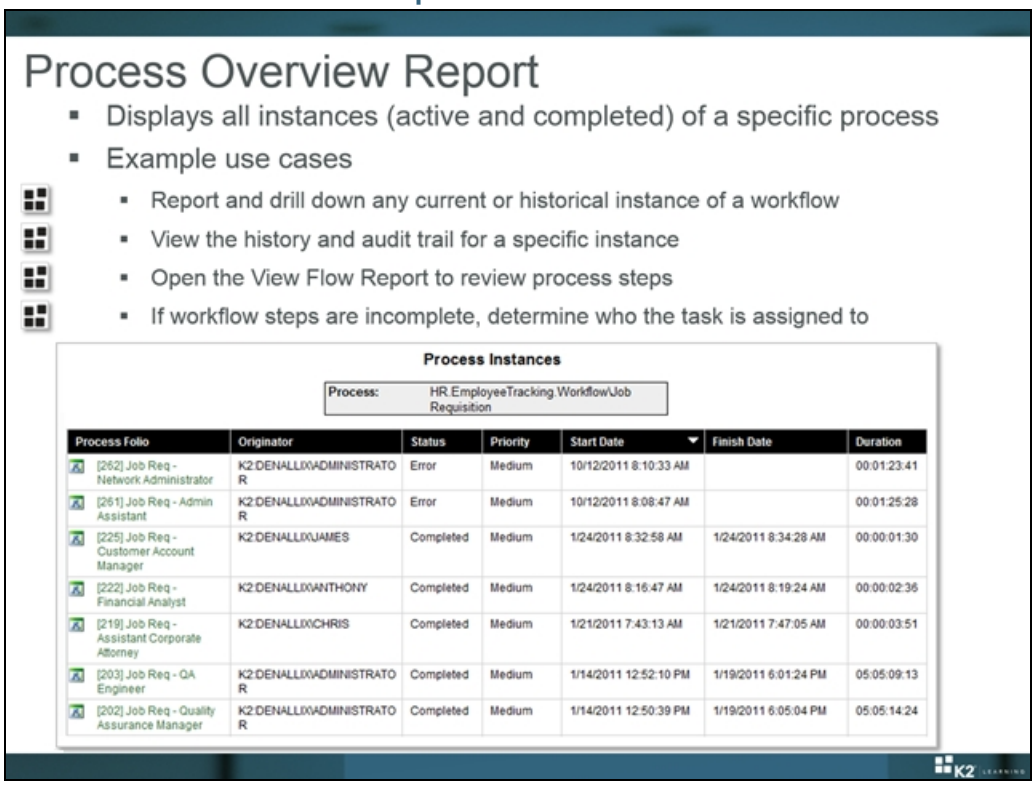
The User Performance report will display, for a specific task in a specific process, how long it takes users to complete the task or how many tasks each user has completed. This report is often used for team supervision and determining average workload and performance reviews. Typical questions that this report can answer include "Over a given time frame, how long does it take each user to complete the same task?" or "How many instances of the same task has each user completed over a given time frame?".

Note that the duration for this report is determined from the time the task is allocated to the user (in other words, when the task appears on their worklist) until they complete it (in other words, they select an action and complete the task). It does not show the time worked on the task, in other words the interval when the user opened the form for the task until they close the form.

Summary

- The User Performance report is used to query the “total number of tasks completed” or “average duration to complete the task”, per user, for a specific task
- The task is counted from the time the task is allocated to the user until they complete it

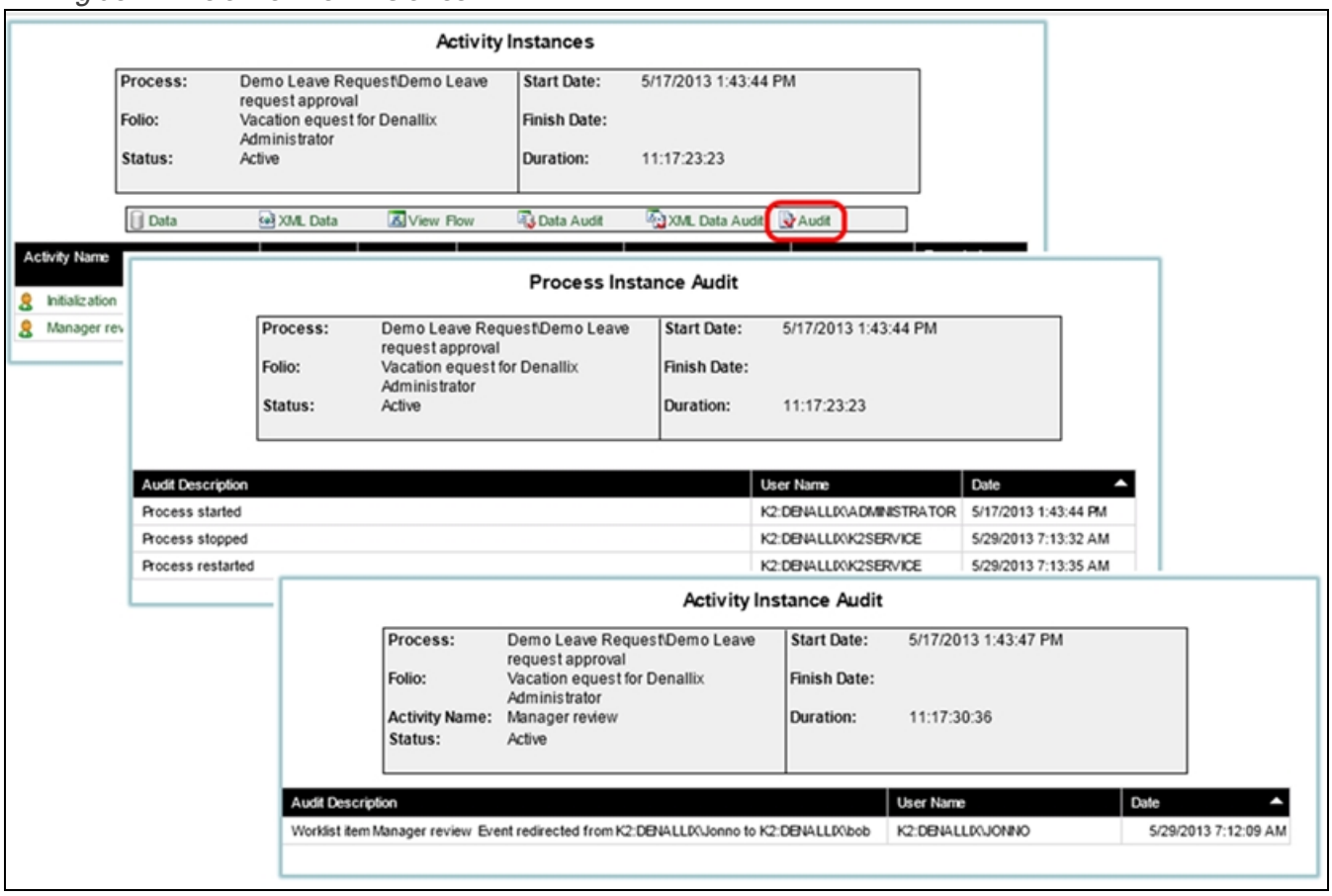
Process Overview Report



Video

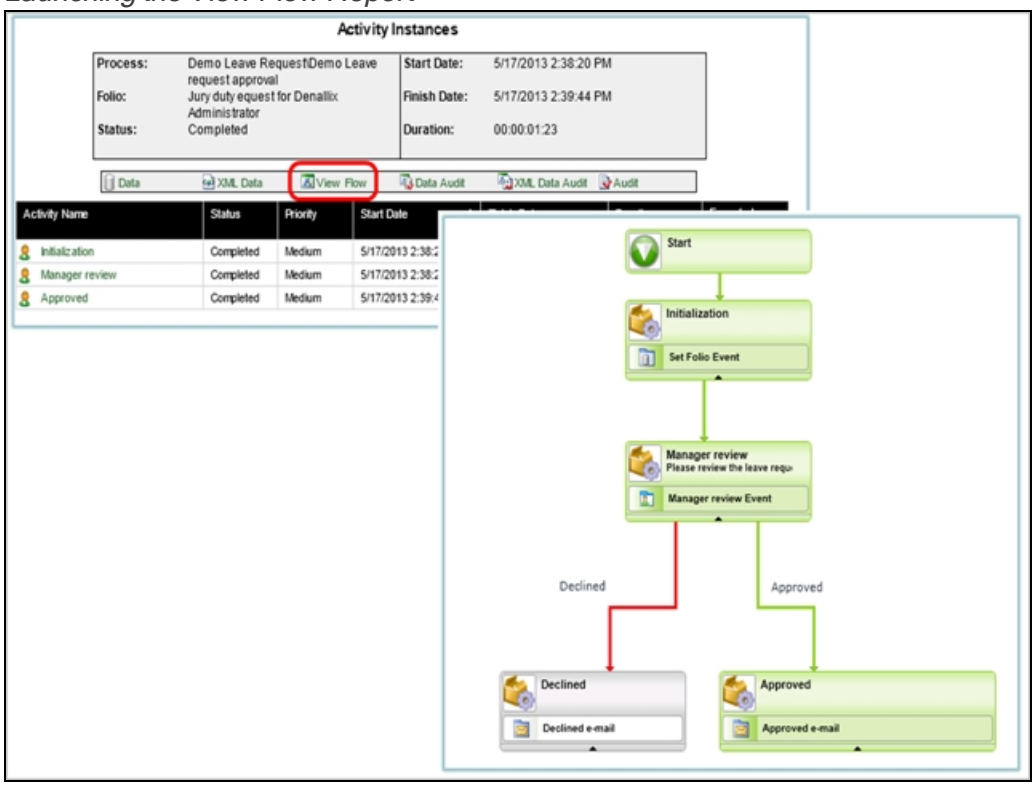
The Process Overview report is probably the most-often used report in K2. This report is used to display detailed information for every single process instance ever executed (or currently being executed) in a K2 environment. The biggest feature of this report is the ability to drill-down into each component for more information, and the link to launch the View Flow report which will display the execution path with additional drill-down for a process instance.

Drilling down into a workflow instance



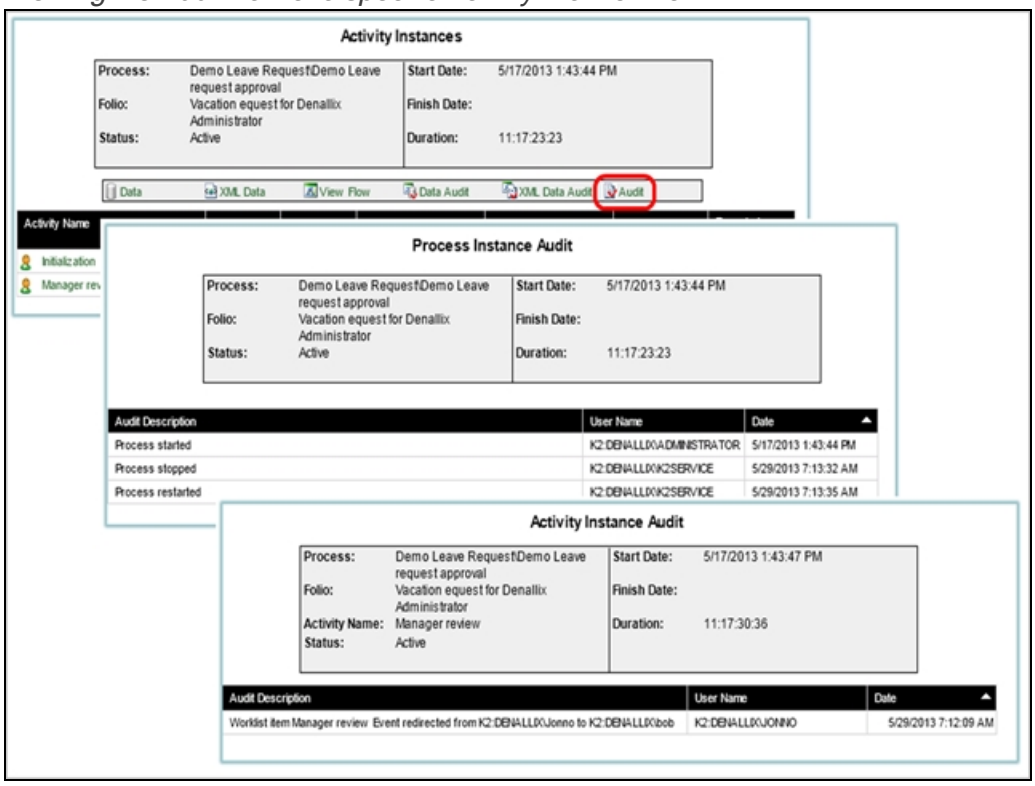
This report is also one of the places where you may launch the View Flow report to view the path that was executed by the workflow instance.

Launching the View Flow Report



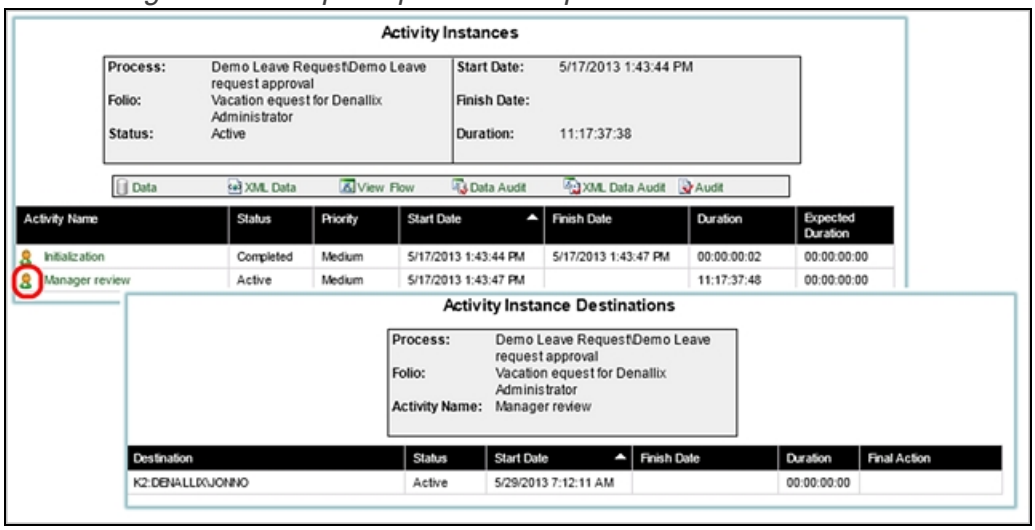
This report exposes workflow history and audit data for all process instances ever executed by the K2 server (excluding process instances that were deleted and the “delete workflow history” option was selected). It is often used to audit specific instances of a workflow, to determine the exact dates that something happens in the workflow and during debugging and troubleshooting to determine where a process might be failing.

Viewing the Audit trail for a specific Activity in a workflow



For activities that are still active, you can use the Process Overview report to determine which users currently own the tasks that still need to be completed. For activities that have already completed, you can use this report to investigate which users completed the task, when they completed it and what their selected Action was.

Determining which users participated in a step in the workflow



Determining what Action a specific user took in the workflow

Activity Instance Destinations

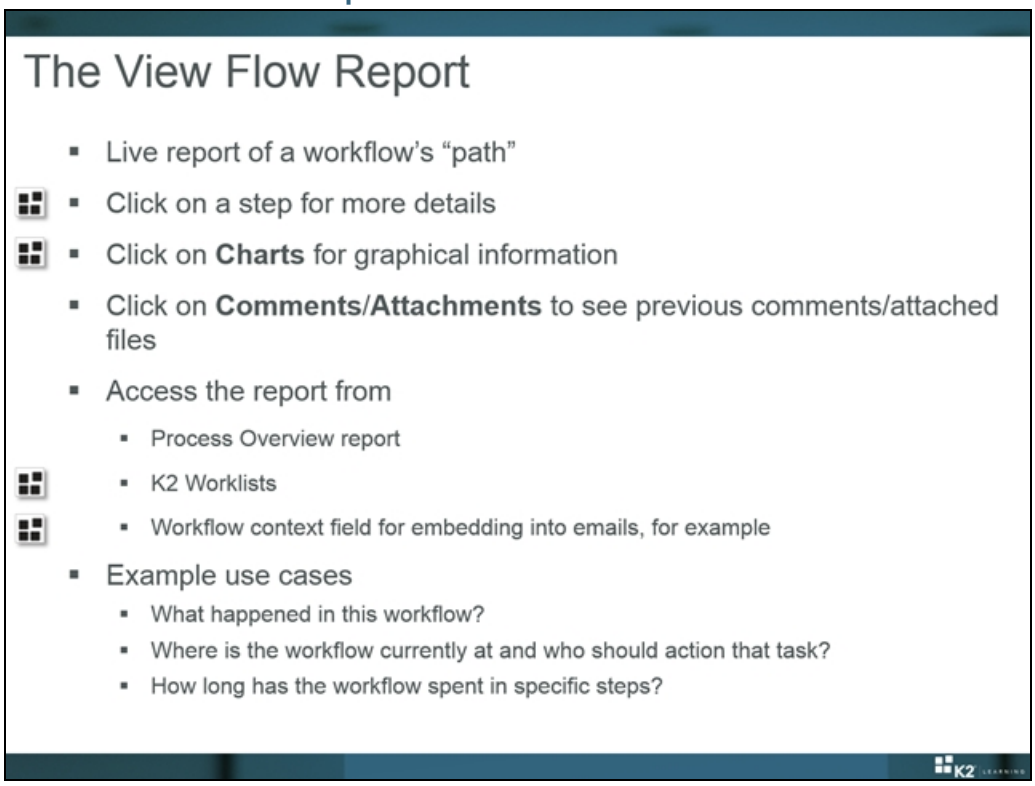
Process: HR.EmployeeTracking.Workflow\Employee Onboarding
Folio: [224] Emp Onboard - Pierce Anderson
Activity Name: HR Approval

Destination	Status	Start Date	Finish Date	Duration	Final Action
K2:DENALLD\JOHN	Completed	1/24/2011 8:23:47 AM	1/24/2011 8:24:35 AM	00:00:00:47	Approve

Summary

- The Process Overview report is probably the most useful report for development, testing, support and administration tasks
- The report shows information for both completed workflows as well as currently-active workflows
- Drill down all the way from a process definition to the individual events in a specific instance of the workflow

The View Flow Report



Video

The View Flow report is a useful tool to investigate the path followed by a particular workflow instance, and to gain more insight into the events that occurred during the process instance. If a process is still active, the View Flow will also update in real time to reflect the current status of the workflow instance.

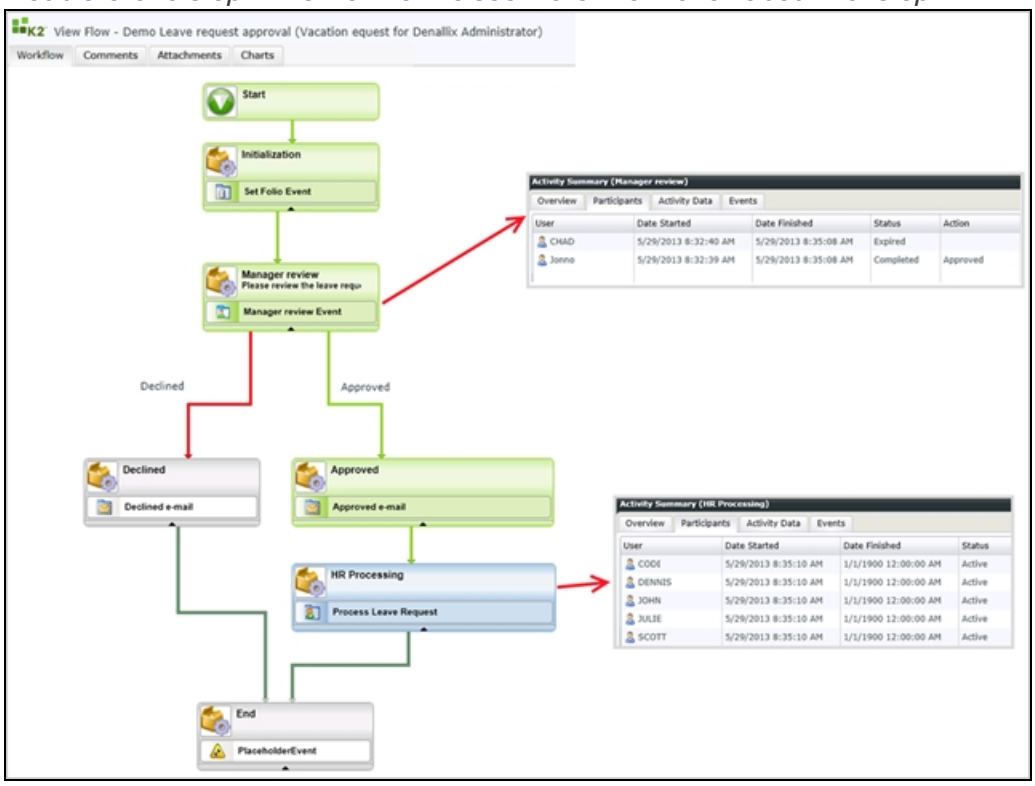
The View Flow report uses color-coding to identify which steps and lines in the workflow were executed. Green items mean that the step was completed or the line was followed, red lines indicate a path that was not followed, blue items are currently active and gray items were not executed or not executed yet.

Note
Users will need at least "View" or "View Participate" permissions on the process definition in order to run the View Flow report for a process instance.

You can use this report in many different ways, such as:

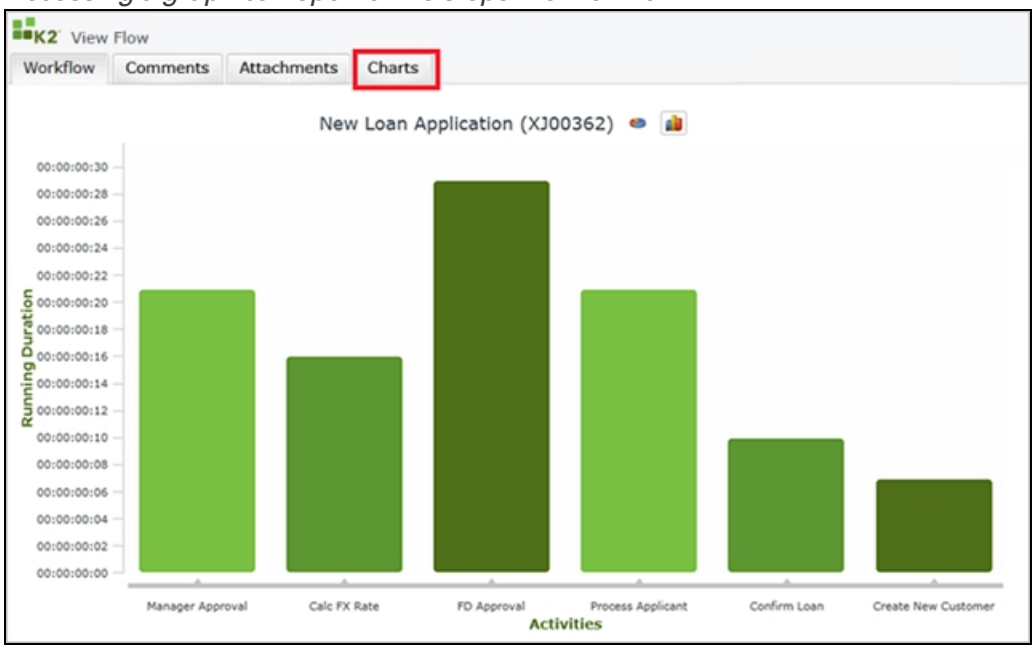
- Seeing the history of a specific workflow
- Drilling down into steps to see who actioned a task, when and what their actions were or determining where a workflow is currently at, and which users need to action a task in the workflow.

Double-click a step in the workflow to see more information about that step



- Viewing process-level data values
- Viewing graphical reports such as how long a workflow has spent in each step

Accessing a graphical report of the steps in a workflow

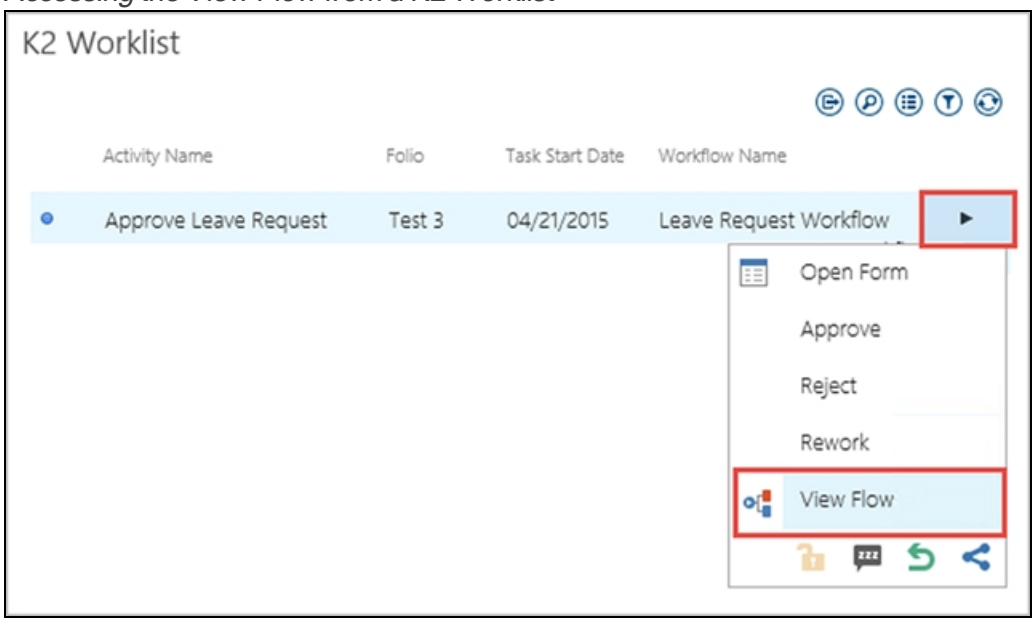


- Viewing the comments and attachments that were added to the workflow through the standard comments/attachments functionality in K2

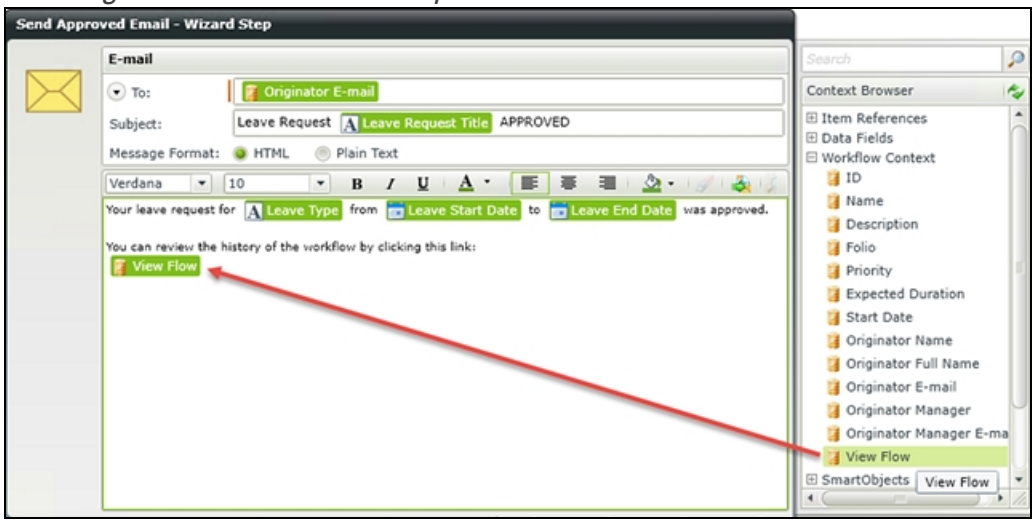
You can access the View Flow from many places, including:

- The Process Overview report

- From K2 Worklists
  Accessing the View Flow from a K2 Worklist



- The workflow designer exposes the View Flow URL so that you can use it in wizards (e.g. emailing a link to the View Flow
  Including a link to the View Flow report from the Context Browser



A common question is whether it is possible to display the View Flow report to external parties (i.e. users outside of the organization's AD domain). This is possible, but you should bear the following points in mind:

- The users must be able to authenticate successfully against K2, and this usually implies a custom security provider and some mechanism to get the user's credentials to the K2 server
- External users need "View" or "View Participate" permissions on the workflow, set up with process permissions
- External users need to install Microsoft Silverlight on their machines to view the report
- The K2 ports (5555 and 5252) will need to be opened on any firewalls to allow the View Flow component to retrieve process information from the K2 server

In certain circumstances, organizations may want to re-host the K2 View Flow report within a custom application, such as an ASP.NET or Windows Forms application, or to add a link to the View Flow report as part of an application. The View Flow component can be re-hosted by constructing a URL that points to the View Flow report, and setting the correct query string parameters for the View Flow URL (the most important parameter being the ProcessId - this is what tells the View Flow report which Process Instance to load).

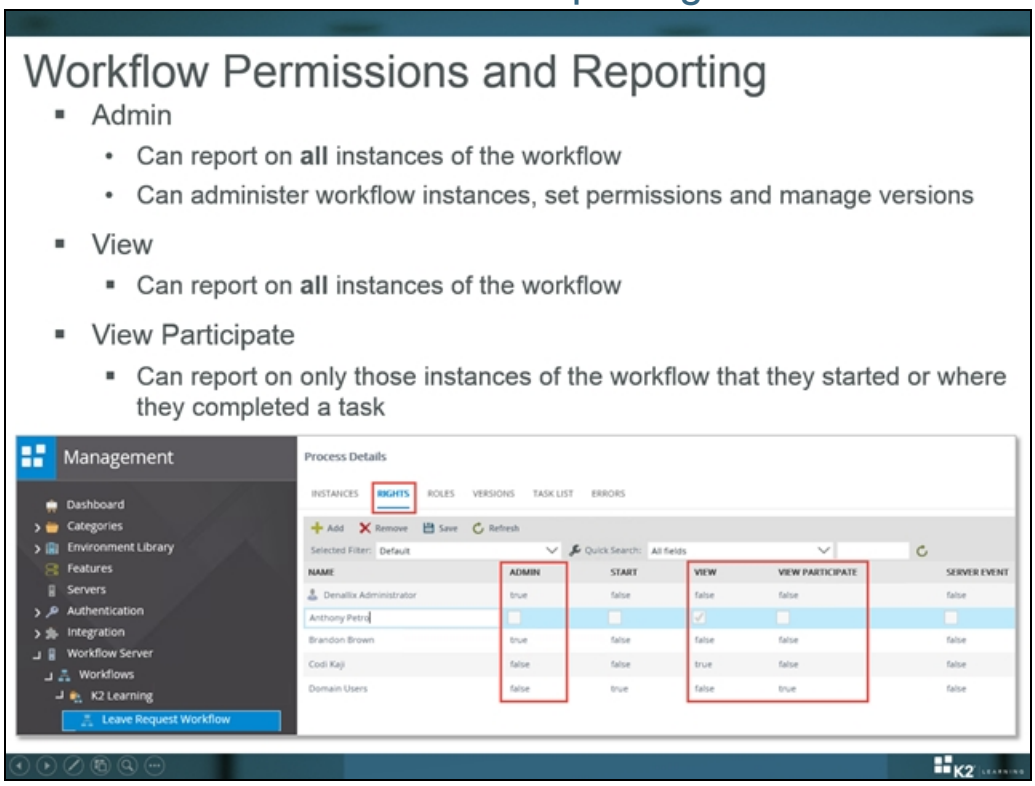
An example of a URL that will display the View Flow report is as follows:
http://-
work-
space.denallix.com/ViewFlow/ViewFlow.aspx?ViewTypeName=ProcessView&K2Server=DLX:5252&HostServerName=DLX

This URL can be broken down into the following parts:
http://WORKSPACEURL/ViewFlow/ViewFlow.aspx?ViewTypeName=ProcessView&K2Server=[K2SERVERNAME]:
[WORKFLOWSERVERPORT]&HostServerName=[K2SERVERNAME]&HostServerPort=[HOSTSERVERPORT]&Pro-
cessID=[PROCESSINSTANCEID]

Summary

- The View Flow is a live, almost real-time reflection of a workflow's "path"
- You can use it in many ways, for instance:
  - Seeing the history of a specific workflow
  - Drilling down into steps to see who actions a task, when and what their actions were
  - Determine where a workflow is currently at, and which users need to action a task in the workflow
  - Viewing process data
  - Viewing graphical reports such as how long a workflow has spent in each step
  - Viewing the comments and attachments that were added to the workflow through the standard com-
    ments/attachments functionality in K2

Workflow Permissions and Reporting



Video

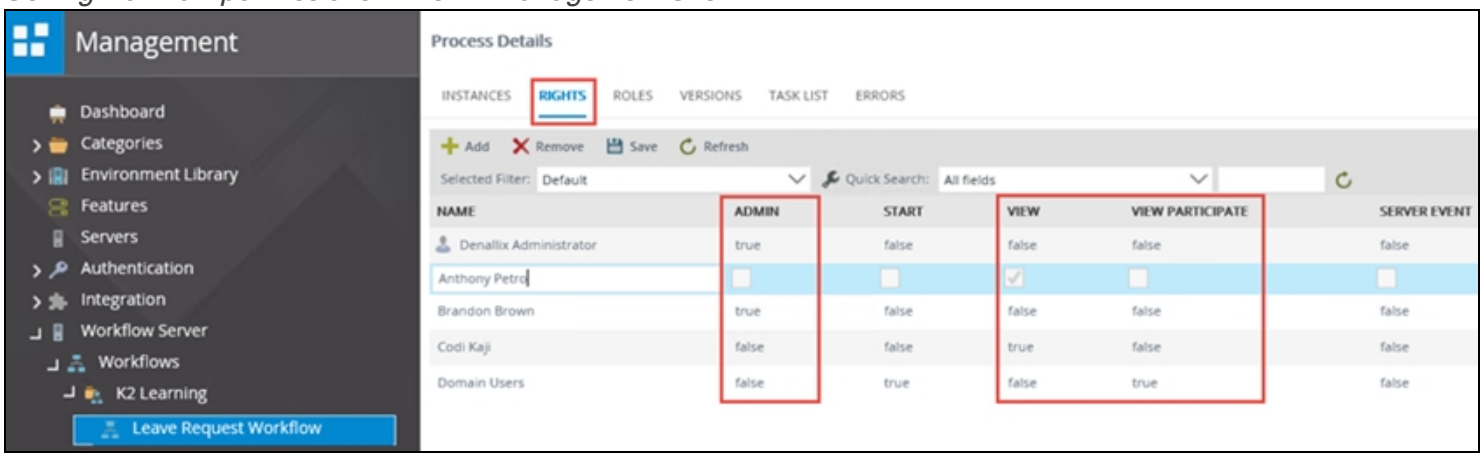
Workflow permissions affect what workflow reporting data is exposed to users. The level of permissions a user or group has on a workflow will determine whether they are able to report on that workflow using the standard K2 reports or retrieve reporting data using the Workflow Reporting SmartObjects. These permissions are set on a workflow definition level using tools like K2 Workspace and the browser-based workflow design tool.

To be able to run reports against workflows, users will require at least one of the following permissions:

- Admin: can report on all instances of that workflow
- View: can report on all instances of that workflow
- View Participate: can only report on those instances that they were specifically involved in ((i.e. they were the originator or they actioned a task.) Being assigned a task does not count, you have to have actioned the task.

By default, the deploying user gains admin rights, but no-one else does, unless configured differently when designing the workflow. Once permissions have been set for a workflow, those permissions apply for all future and past versions of that workflow.

Setting workflow permissions in the K2 Management Site

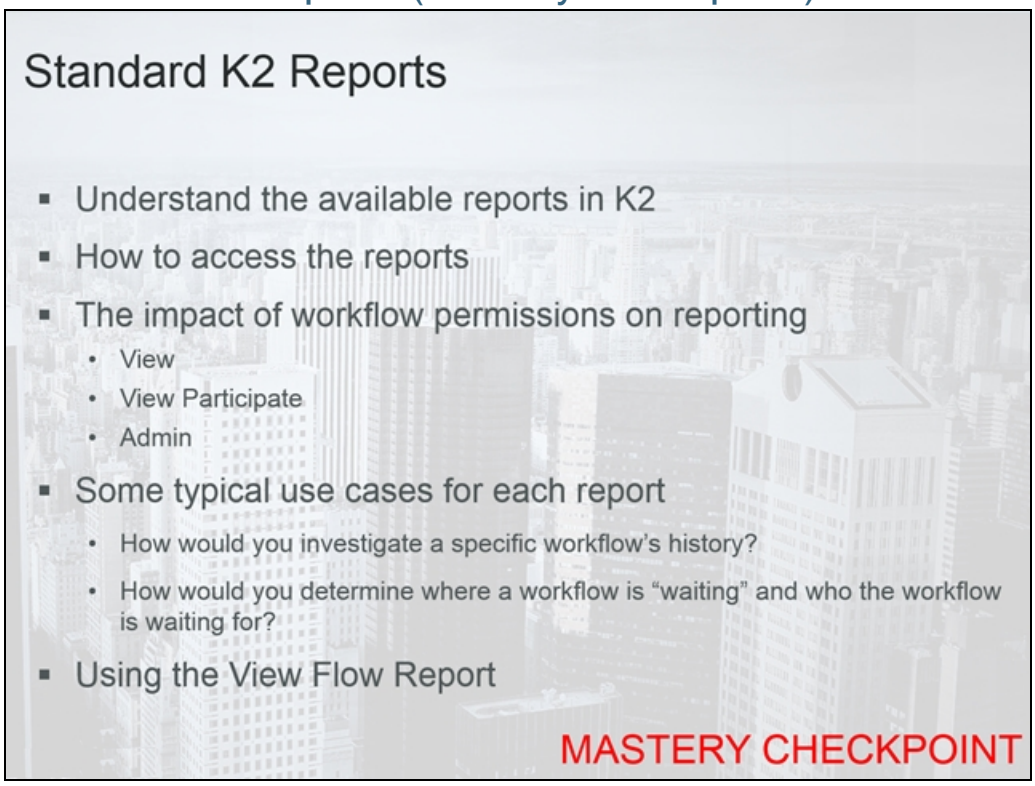


Tip
Workflow permissions are aggregative (not subtractive) and are based on the most-permissive approach (as opposed to most-restrictive).
This means if a user has both View and View Participate permissions, View will take precedence because it is a more permissive right. If a user has both Start and View permissions, they will be able to both start the workflow and view all instances of the workflow in the K2 reports.

Summary

- Workflow permissions affect the workflow reporting data that is exposed to users
- Admin or View rights can report on all instances of that workflow
- View Participate can only report on those instances that they were specifically involved in (i.e. they were the originator or they actioned a task)
- Workflow permissions can be set in K2 Workspace, K2 process portals and in some workflow design tools

Standard K2 Reports (Mastery Checkpoint)



Video

This is a checkpoint for the information covered in Part 1 of this module: Standard Reports in K2. If you are attending a training session led by a K2 instructor, this is your opportunity to ask questions on any topics you might not yet understand.

These are the main concepts you should understand:

- What standard reports are available in K2 and where to access them
- How to use the standard reports to answer common questions
- How to use the View Flow report
- How workflow permissions affect reporting

Knowledge-check questions

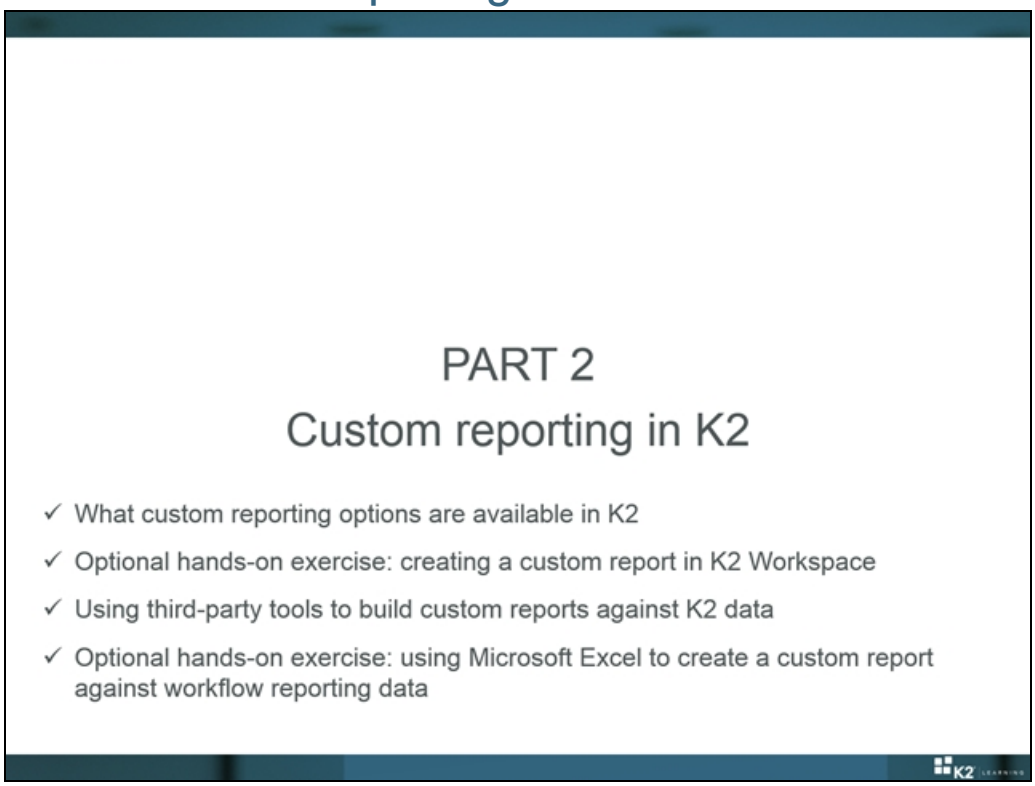
Q: How would you investigate a specific workflow's history?
Reveal answer
A: Using the View Flow report or the Process Overview report.

Q: How would you determine where a workflow is "waiting" and who the workflow is waiting for?
Reveal answer**A:** Use the View Flow and drill down into a specific step, or use the Process Overview report and drill down into a specific step.

Q: Which report would you use to determine who your star performers are for a specific workflow task, in terms of tasks completed?
Reveal answer**A:** The User Performance report.

Q: Which report will help you determine which activity might be the bottleneck in a particular workflow design?
Reveal answer**A:** The Activity Statistics report.

Part 2: Custom reporting in K2

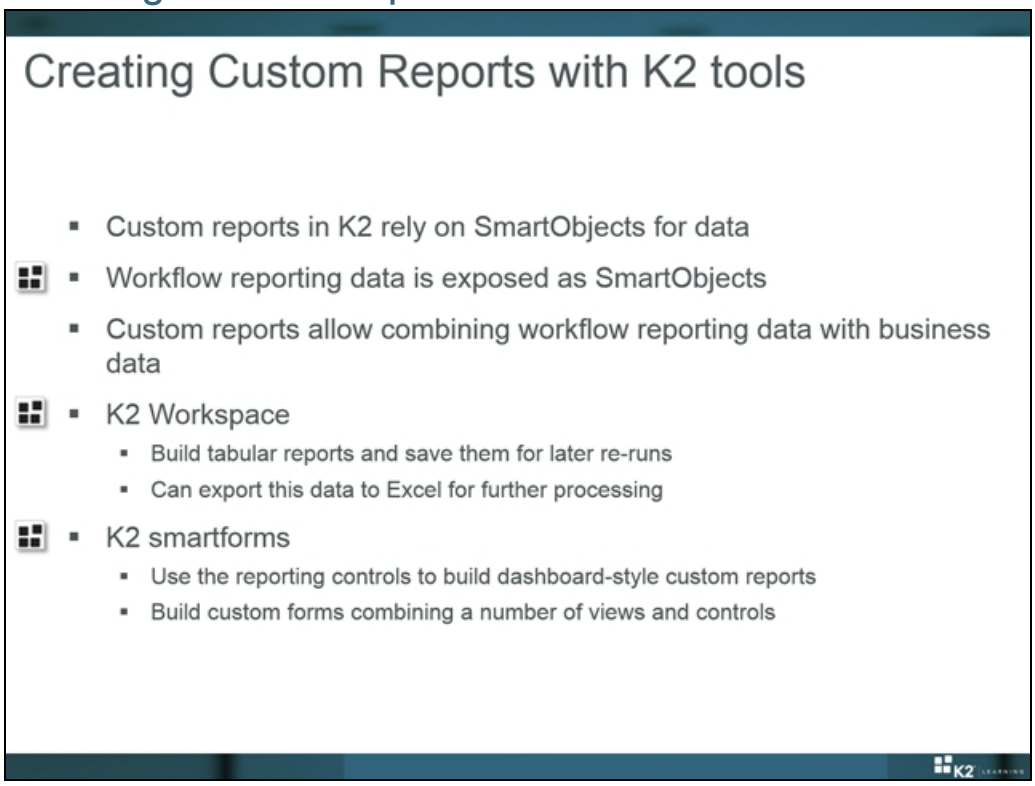


Video

In Part 2 we will look at the custom reporting options that are available in K2: what tools K2 provides for building custom reports (followed by an optional exercise to build a custom report in K2 Workspace) and how to use third-party tools to build custom reports against K2 data (followed by an optional exercise to build a custom report with Microsoft Excel).

Note
If your organization will not be utilizing any custom reporting on K2, you may skip Part 2 of this course module.

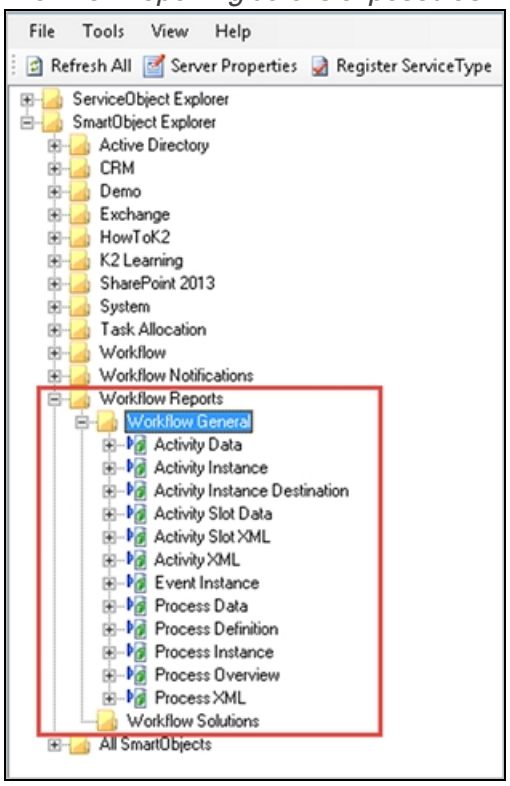
Creating Custom Reports with K2 tools



Video

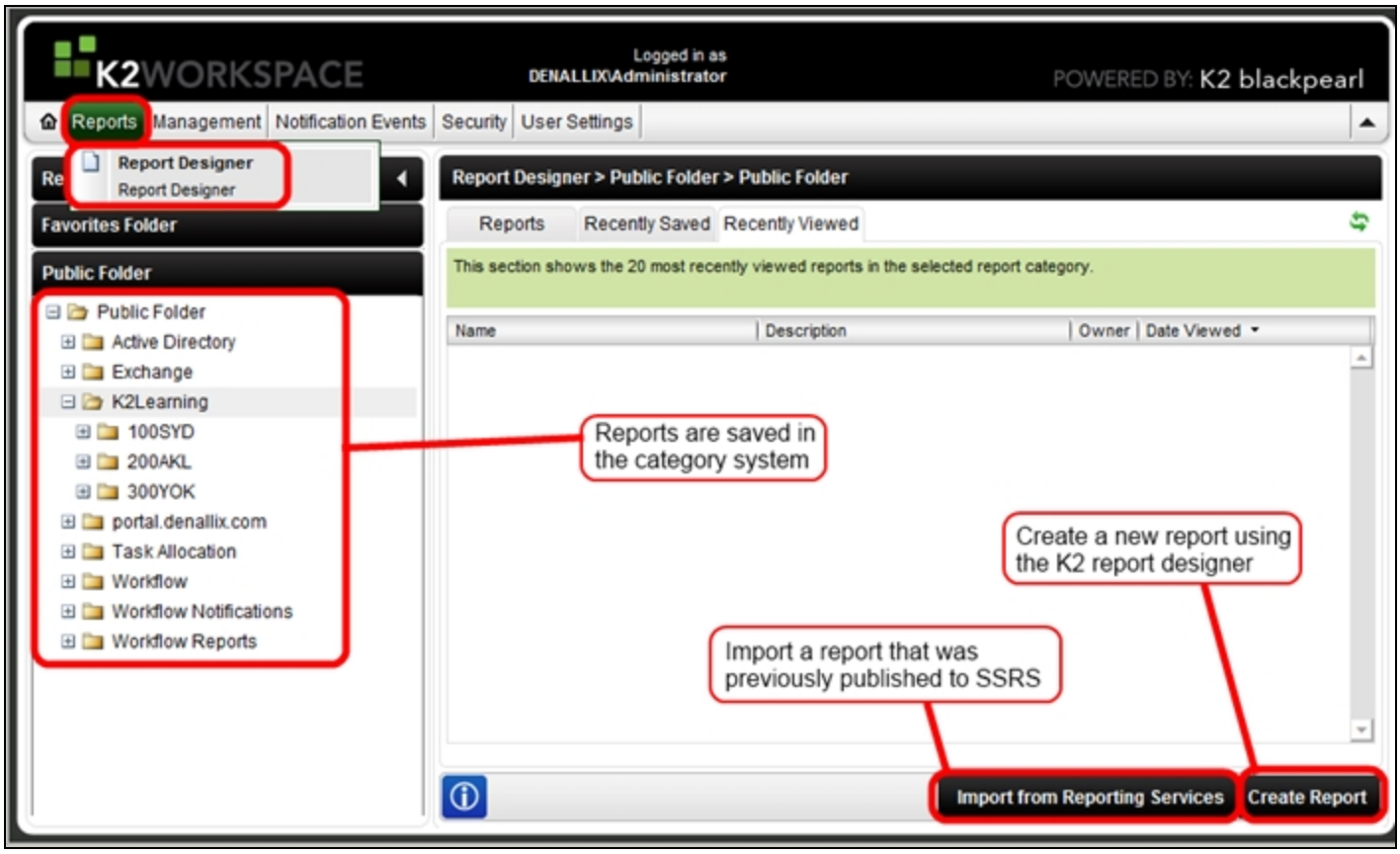
A common requirement when building workflow-based applications, is the need to combine workflow reporting data with business data. K2 provides custom reporting tools that allow you to use SmartObjects to build custom reports. Because workflow reporting data is exposed as SmartObjects, you can use these tools to build custom reports against workflow reporting data as well as other SmartObjects, which allow you to build reports that combine business data with workflow data. Here is a common use case: suppose you have an Expense Claim Approval process, where the expense claim data is stored in an external database while the process is based on a K2 workflow process. Because both systems can be exposed as K2 SmartObjects, it is possible to create reports that combine data from both the workflow and the external database to create a custom report that shows the total amount of expense claims that are currently at the “Finance Approval” stage.

Workflow reporting data is exposed as K2 SmartObjects

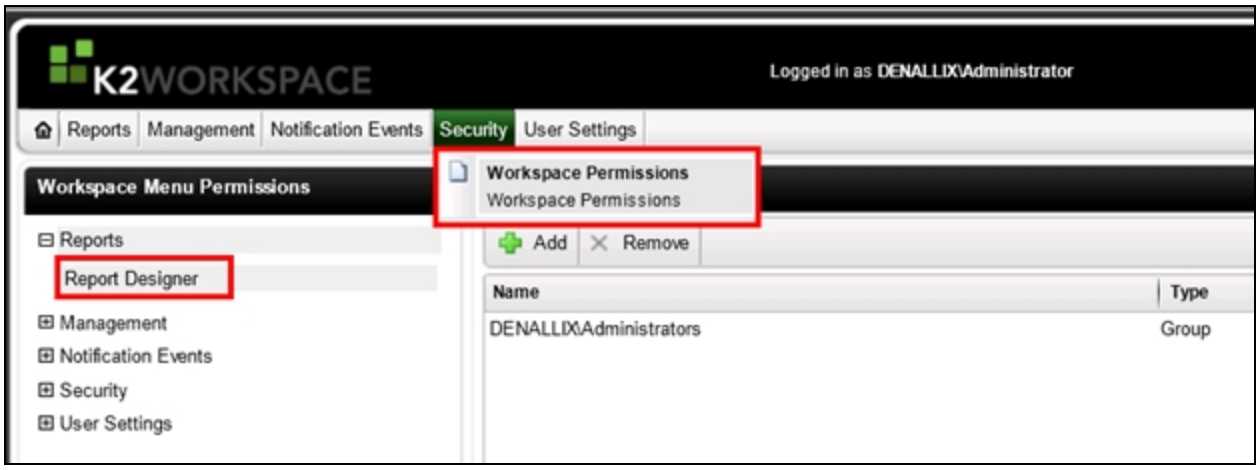


The two tools K2 provides to build custom reports are the Report Designer in K2 Workspace and the ability to build custom reports with SmartForms, using the available reporting controls.

The K2 Workspace Report Designer allows users with appropriate permissions to design custom Tabular, Matrix or Summary reports using a web-based design environment. (It is not possible to create graphical reports in this design tool.) An additional feature is the ability to import a report that was previously deployed to a SSRS server. (This option is only available if the K2 installation was associated with a SSRS server when K2 was installed.) This makes it possible for end users to run SSRS reports from the same reporting environment as they would use for running reports designed with the K2 report design tool. You can also export existing report definitions from K2 into an .rdl file for further design and development in a SSRS design tool like Visual Studio.



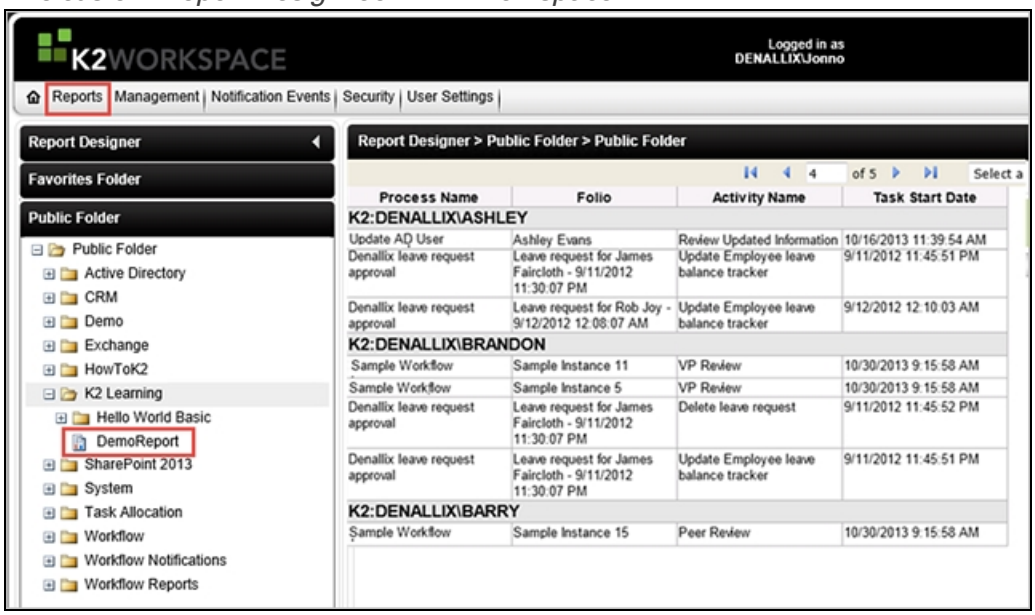
Administrators can restrict who may design and run reports in this environment by setting up security in K2 workspace, as shown below:



When creating a new report, designers can select data from any SmartObjects published on the K2 environment. If SmartObject Associations have been defined, K2 will automatically allow the user to combine data from additional SmartObjects based on the associated property. As part of the report design you can optionally summarize, group and filter data, and apply styling to the report. When the report is saved, it will be saved in a specified folder in the Category System, and other users will then be able to re-run the report at any time in the future.

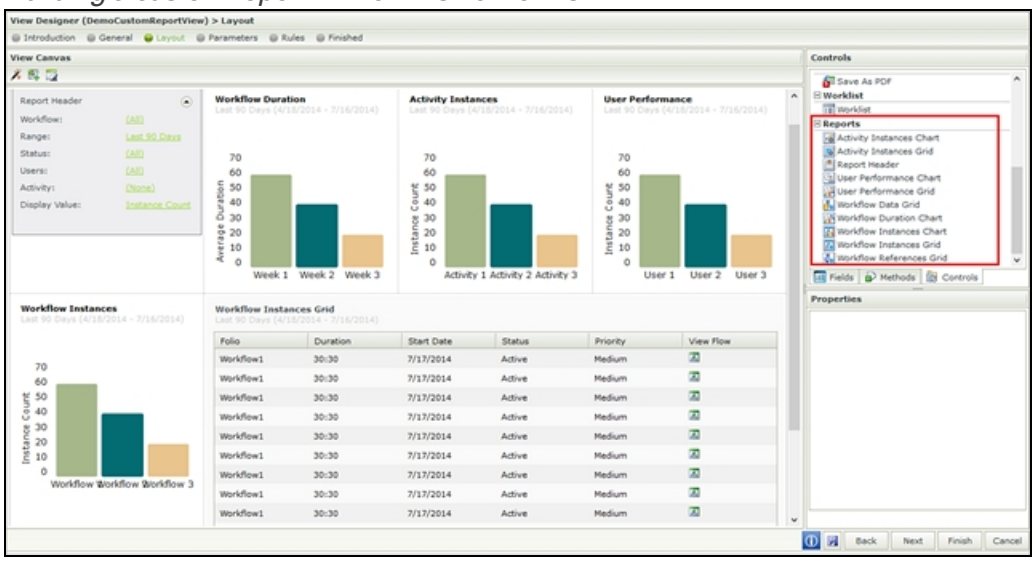
Note
This module will focus more on the K2 Workspace report design tool, since the SmartForms training course will go into more detail about building Forms and Views with Controls and Rules. Those principles apply equally when you use SmartForms to build custom reports.

The custom Report Design tool in K2 Workspace



When building custom reports with SmartForms, you can use the available Reporting controls and all the power of K2 smartforms to build interactive reports that respond to user input and selections.

Building a custom report with a K2 smartforms

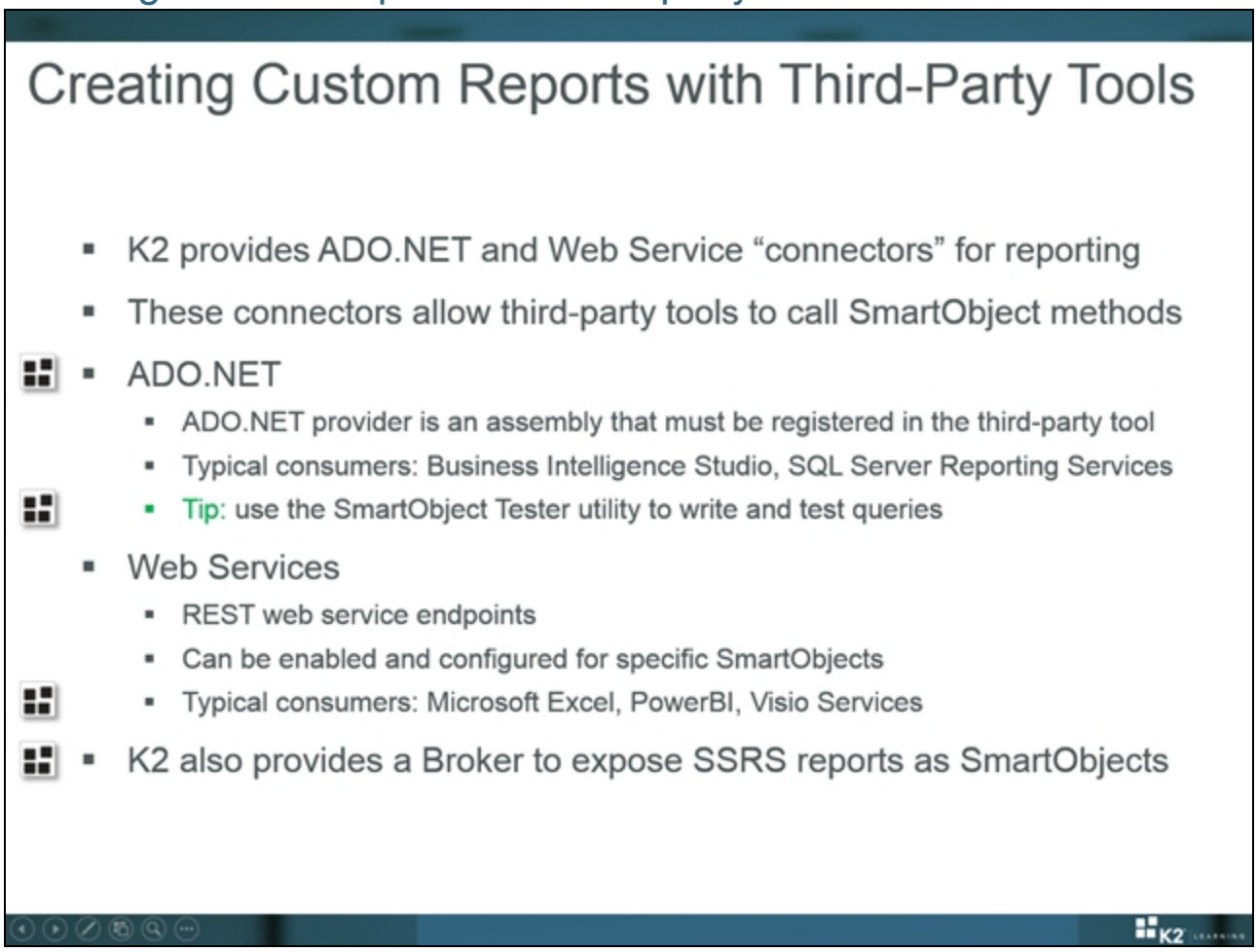


Note
Because the custom reporting tools use different formats, it is not possible to migrate reports built in Workspace to SmartForms; you have to rebuild them.

Summary

- K2 provides two custom reporting tools that allow you to use SmartObjects to build custom reports
- You can use these tools to build custom reports against workflow reporting data as well as other SmartObjects
- There are two custom reporting tools:
  - The Report Designer in K2 Workspace
    - Build tabular style reports
    - Can export data to Excel for further processing
    - Can save reports for later re-runs and can edit those reports as well
  - Build custom reports with SmartForms
    - Use the available reporting controls to report on workflow data
    - Use other controls to report on SmartObject data

Creating Custom Reports with third-party tools

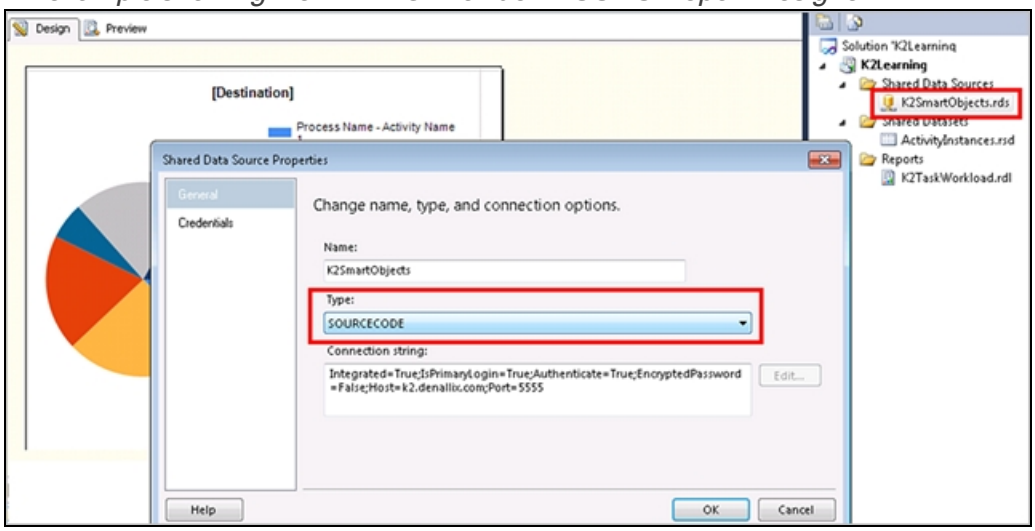


Video

It is possible to use third-party applications like Microsoft Excel or Microsoft Visio or report design tools like SQL Server Reporting services to build custom reports against K2 reporting data, or more accurately against data exposed as SmartObjects. The main consideration is whether the third-party tool is able to consume either the K2 APIs or the K2 web services. An example of an API is the ADO.NET provider, which is an API that allows you to write SQL-like queries against SmartObjects. K2 also provides services like the REST web services that can expose SmartObjects as web method calls.

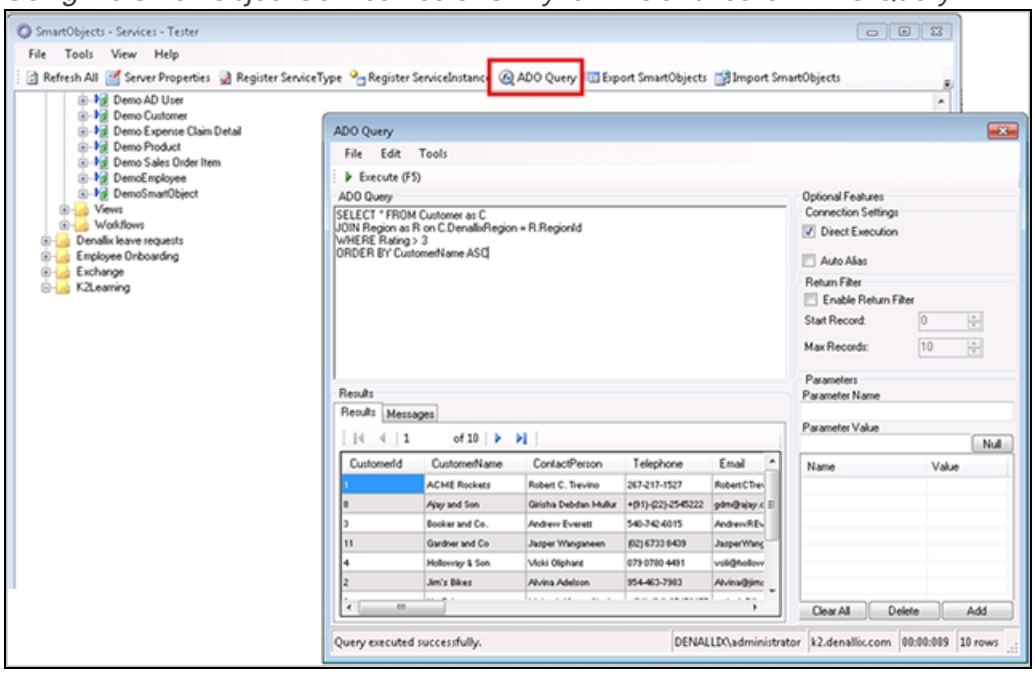
The ADO.NET provider must be registered in the tool (for example, in SSRS it must be added to the SSRS configuration files and copied to the SSRS server). One of the biggest benefits of this approach is that it allows you to combine results from multiple SmartObjects with JOIN and UNION statements, regardless of where the underlying data resides.

An example showing the K2 ADO Provider in SSRS Report Designer



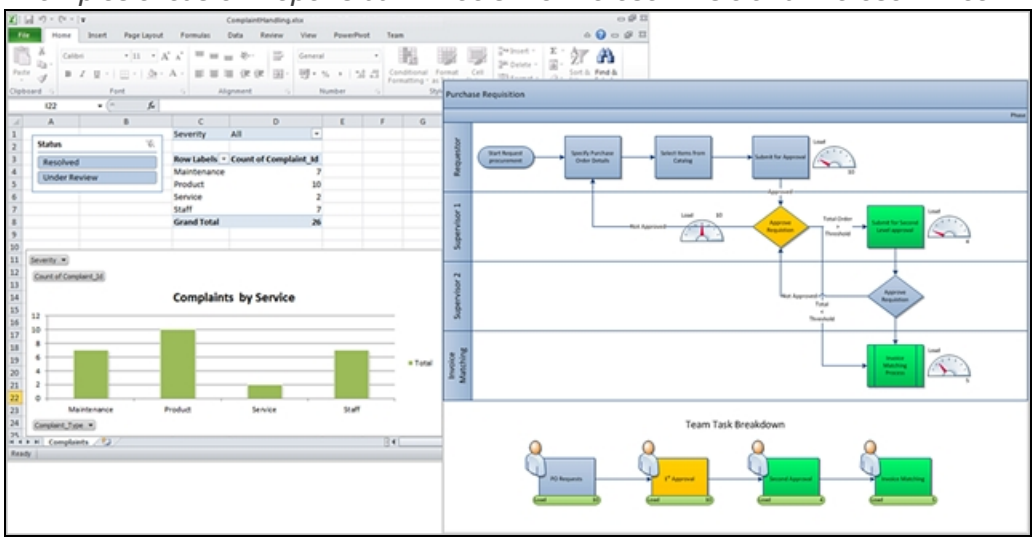
Tip
When using the ADO API you will write ADO queries (e.g. "SELECT * FROM Employee), and we recommend using the SmartObject Service Tester utility to write and test queries before implementing the queries in the reporting tool.

Using the SmartObject Service Tester Utility to write and test an ADO Query



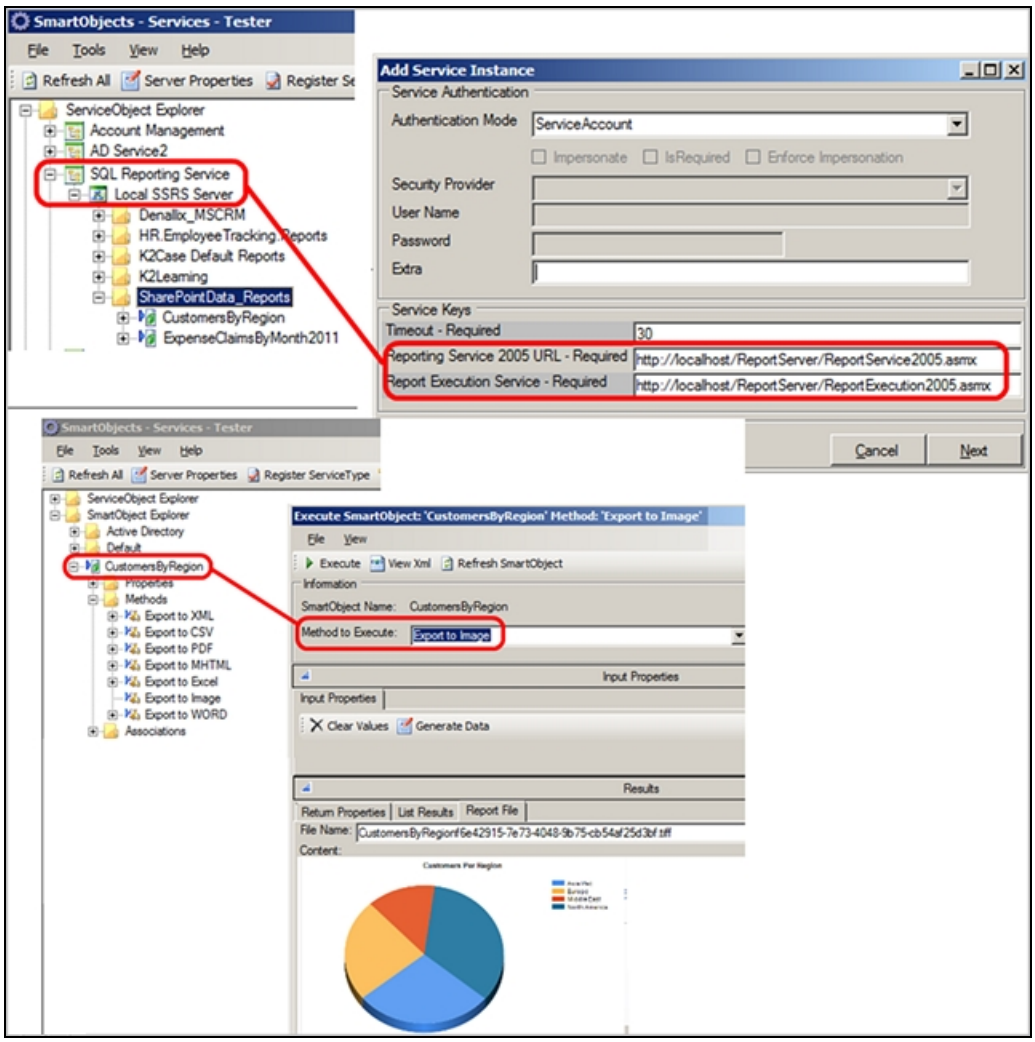
If you will be using the REST services, you may need to enable the endpoints first. You can also optionally specify only certain SmartObjects that must be exposed as service endpoints. (The REST endpoints are configured by editing the K2 Host Server configuration file). Typical consumers of these service endpoints are office apps like Excel, PowerPivot, PowerBI, and Visio services. This feature allows you to build custom reports or other interfaces that represent the data returned by SmartObjects. For example, to create custom reports against workflow reporting data, or creating custom reports that expose data stored in some back-end system, if that data is exposed as a SmartObject

Examples of custom reports built in tools like Microsoft Visio and Microsoft Excel



Finally, it is worthwhile to know that K2 provides a Service Broker that allows you to expose SSRS reports as SmartObjects. This allows you to do things like generating a report from SSRS, export it to a PDF file and then attaching the PDF file to an email as part of a workflow.

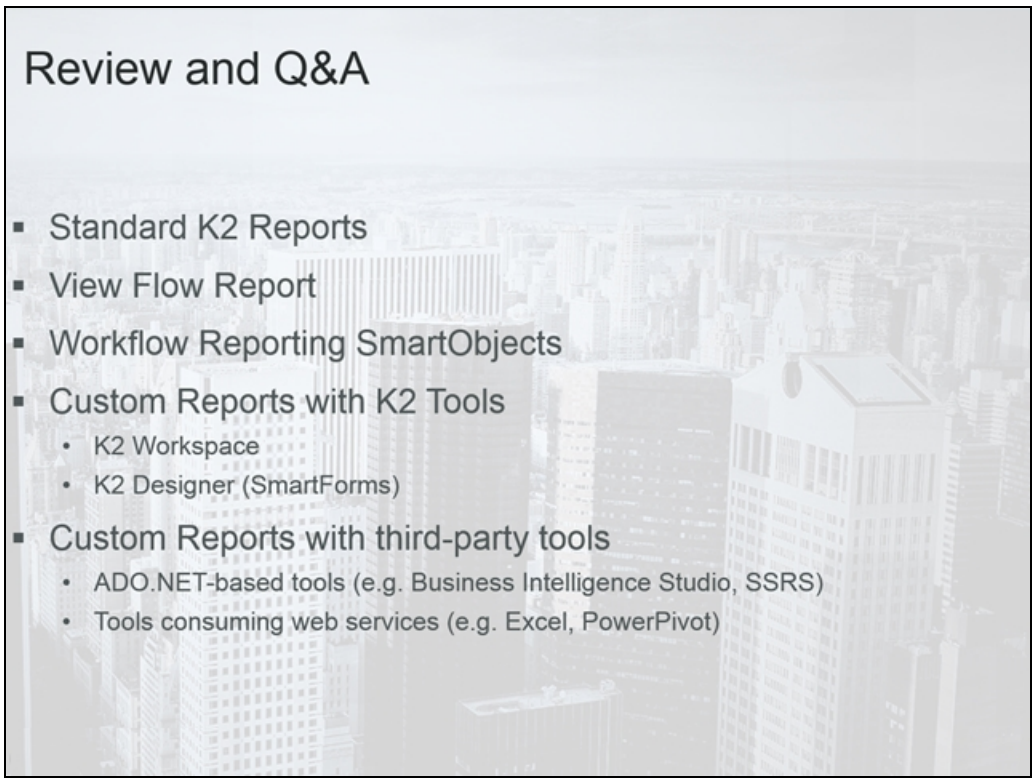
The SSRS Service Broker



Summary

- K2 provides two methods to consume SmartObjects in third-party reporting tools:
  - ADO.NET (this is an assembly that allows the running of SQL-like queries against SmartObjects)
  - Web Services (these are REST web services that allow tools to call a service endpoint to retrieve data)
- ADO.NET key points:
  - ADO.NET provider must be registered in the tool
  - Use the SmartObject Service Tester utility to write and test queries
  - Can combine results from multiple SmartObjects with JOIN and UNION statements
- Web Services key points:
  - REST endpoints that expose specific SmartObjects methods
  - Enable and configure the REST endpoints by editing the K2 Host Server config file
- K2 provides a service broker that allows you to expose SSRS reports as SmartObjects

Review and Q&A



Video

This topic is just a summary of the information covered in this module: Introduction to K2 Applications with K2 Designer. If you are attending a training session led by a K2 instructor, this is your opportunity to ask questions or, if time permits, discuss your own plans for using K2 in your environment.

In summary, here is what we covered in this training module:

- The Standard Reports in K2 and how to use them
- Using the View Flow report and clicking on steps to drill down
- How to build Custom Reports in K2 workspace
- Using third-party tools to build custom reports
- How to build a custom report in something like Excel

Knowledge-check questions

Q: Which reports do you think would be most useful to you, and why?
A: (discussion question)

Q: What kind of custom reporting tools are you using today?
A: (discussion question)

Q: If you were to build custom reports against K2 data, which tool would you use and why?
A: (discussion question)

Terms Of Use

ACCEPTANCE OF TERMS

THIS DOCUMENTATION IS SUBJECT TO THE FOLLOWING TERMS OF USE ("TOU"). SOURCECODE TECHNOLOGY HOLDINGS INC. ("SOURCECODE") RESERVES THE RIGHT TO UPDATE THE TOU AT ANY TIME WITHOUT NOTICE. THE MOST CURRENT VERSION OF THE TOU CAN BE REVIEWED BY CLICKING ON THE "TERMS OF USE" LINK IN THE TABLE OF CONTENTS.

COPYRIGHT

THIRD-PARTY INTEGRATION

SOURCECODE MAY MODIFY, ADJUST OR REMOVE FUNCTIONALITY IN THE K2 SOFTWARE IN RESPONSE TO CHANGES MADE TO VERSIONS OF THE RELEVANT MICROSOFT OR OTHER THIRD PARTY PRODUCTS. THIS DOCUMENTATION MAY PROVIDE ACCESS TO OR INFORMATION ON CONTENT, PRODUCTS, AND SERVICES FROM THIRD PARTIES. SOURCECODE TECHNOLOGY HOLDINGS INC. IS NOT RESPONSIBLE FOR AND EXPRESSLY DISCLAIMS ALL WARRANTIES OF ANY KIND WITH RESPECT TO THIRD-PARTY CONTENT, PRODUCTS, AND SERVICES.

NOTICE REGARDING CONTENT OF THIS DOCUMENTATION

THE INFORMATION CONTAINED IN THIS DOCUMENT AND ITS ASSOCIATED RESOURCES, INCLUDING UNIFORM RESOURCE LOCATORS AND IDENTIFIERS, IS SUBJECT TO CHANGE WITHOUT NOTICE.

UNLESS EXPLICITLY STATED OTHERWISE, THE PEOPLE, ORGANIZATIONS, COMPANIES, PLACES, DOMAIN NAMES, E-MAIL ADDRESSES, PRODUCTS, LOGOS, AND EVENTS DEPICTED ARE FICTITIOUS AND NO ASSOCIATION WITH ANY ACTUAL PEOPLE, ORGANIZATIONS, COMPANIES, PLACES, DOMAIN NAMES, E-MAIL ADDRESSES, PRODUCTS, LOGOS, AND EVENTS IS INTENDED, OR SHOULD BE INFERRED UNDER ANY CIRCUMSTANCE.

WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS DOCUMENTATION, THE DOCUMENTS AND RELATED GRAPHICS CONTAINED IN THIS DOCUMENTATION COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. SOURCECODE TECHNOLOGY HOLDINGS INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE DOCUMENTATION AT ANY TIME AND WITHOUT NOTICE. ERRORS AND INACCURACIES MAY BE REPORTED IN WRITING TO DOCUMENTATION@K2.COM