# Adobe® LiveCycle® Enterprise Suite 2 (ES2)

**Table of contents**

Enterprises face a challenge: to more effectively engage with customers beyond the firewall while making efficient use of back-end resources. IT departments need to make investment choices that allow them to better organize and use distributed capabilities. These capabilities must work in concert with scalable applications that engage customers through the appropriate and targeted use of rich and compelling interfaces.

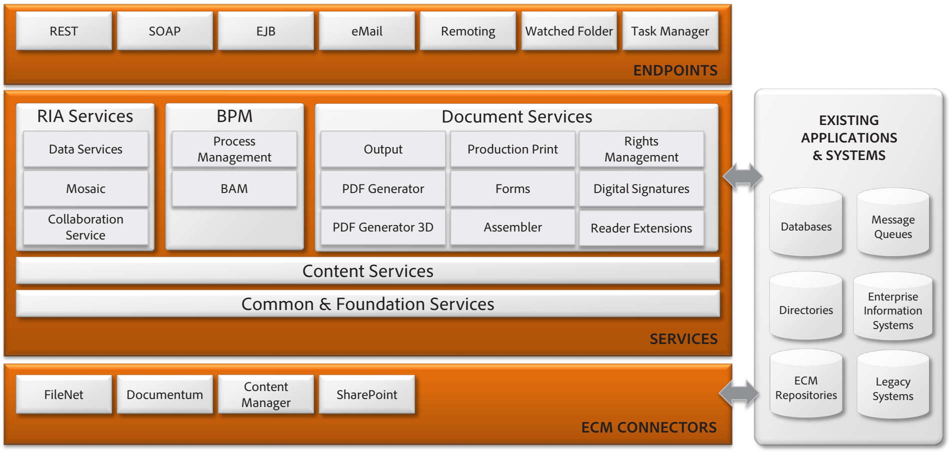
Effectively reaching and engaging customers against a backdrop of rapidly changing business environments, while reducing development costs and maximizing return on investment, demands a service-oriented approach to application delivery. This type of delivery allows back-end processes to be exposed as loosely coupled services and easily assembled into targeted solutions. Applications must be presented to customers through rich and intuitive interfaces that increase user satisfaction, improve process utilization and completion rates, and increase transaction size.

Adobe LiveCycle ES2 exposes software as services and enables effective customer engagement applications that meet these challenges by allowing organizations to quickly assemble more secure, feature-rich solutions that leverage Adobe® PDF and Adobe® Flex technologies to dramatically improve user engagement via the cross-platform Adobe Reader® and Adobe Flash® software installed on over 98% of internet-enabled PCs worldwide. Organizations can choose the appropriate combinations of standards-based user interface technologies, including Flex, PDF, XML, HTML, and Ajax, to most effectively engage their users, drive flexibility, and reduce custom implementations and vendor lock-in.

This paper discusses the Adobe LiveCycle ES2 architecture, providing an overview of how LiveCycle ES2 employs service-oriented principles to enable a more rapid development experience for building scalable applications. By leveraging Adobe Flash, PDF, and Adobe AIR™ within the context of the Adobe LiveCycle ES2 service-oriented architecture (SOA), organizations can engage with their various constituents, such as customers, partners, contractors, to a degree not previously possible. This paper also discusses the types of solutions LiveCycle ES2 provides, including intuitive user experience, business process automation and communication management.

## LiveCycle ES2 architecture overview

The LiveCycle ES2 architecture encompasses clients, user interface technologies, development tools, and foundation infrastructure. The following sections discuss these themes in the context of designing, running, integrating, and administering customer engagement applications.



The LiveCycle ES2 architecture

The LiveCycle ES2 design and development tools work within the application model to coordinate shared assets and dependencies, streamlining the development process and allowing multiple users to work on the same application simultaneously. Design-time revision control, security, and auditing components provide administrators with the tools they need to ensure end-to-end management of the application development process.

LiveCycle ES2 solution modules provide reusable services that can be easily assembled into customer engagement applications. These modules provide wide-ranging and robust support for creating rich internet applications, automating business processes and improve communications.

LiveCycle Content Services ES2 offers comprehensive library services that can be used in conjunction with other LiveCycle modules. This module provides the ability to manage and integrate content into enterprise business processes, streamline content reviews, archive content according to corporate retention policies, and classify content to rapidly locate critical business data.

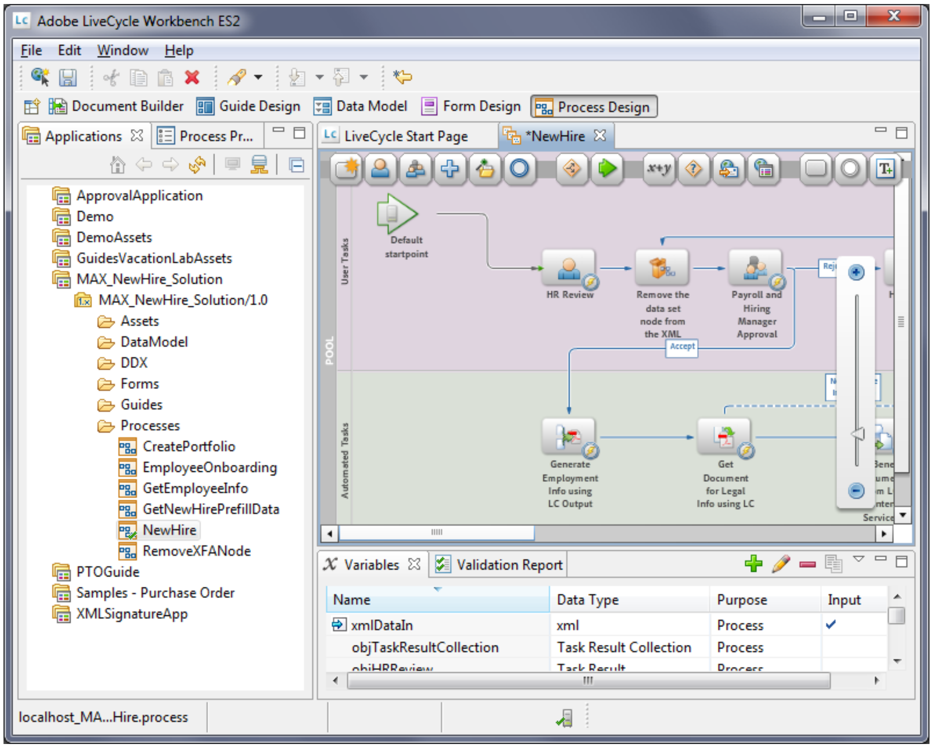
LiveCycle ES2 Connectors for ECM are designed to help integrate LiveCycle applications with industry-leading enterprise content management systems such as EMC® Documentum®, IBM® FileNet, IBM Content Manager and Microsoft® Office Sharepoint® Server.

Adobe LiveCycle Foundation provides an integrated environment that enables the deployment, execution, and management of customer engagement applications. Integral to the Foundation is the Service Container, which provides a common runtime environment for all solution modules and associated services. LiveCycle Foundation services provide facilities that enable solutions to integrate with common IT infrastructures, such as web services, LDAP, JDBC, and FTP.

LiveCycle ES2 includes centralized tools that simplify a wide range of administrative tasks, including configuration, administration, deployment, and monitoring. Each solution module can be independently administered from a single web interface.

## Designing engagement applications

LiveCycle Workbench™ ES2 and Adobe® Flash Builder™ provide integrated form, process, and rich internet application (RIA) design tools that expose the appropriate functionality to specific team members. For example, form designers and business analysts can work collaboratively within LiveCycle Workbench ES2. Developers can extend application functionality through data integration, scripting, and extension of module capabilities through the LiveCycle ES2 software development kit (SDK). Business analysts can define conceptual processes while form designers can reuse form fragments and script components created by developers. Reusable content components (form templates, service definitions, process definitions, policies, DDX files, XML schemas, and images, for example) and form fragments (headers and footers that can be reused across multiple forms, for example) are stored, version-controlled, and protected in LiveCycle ES2. This approach greatly simplifies the maintenance and update tasks related to application development. The LiveCycle ES2 repository can be seamlessly extended with enterprise content management (ECM) systems, and assets can be protected using access control lists.



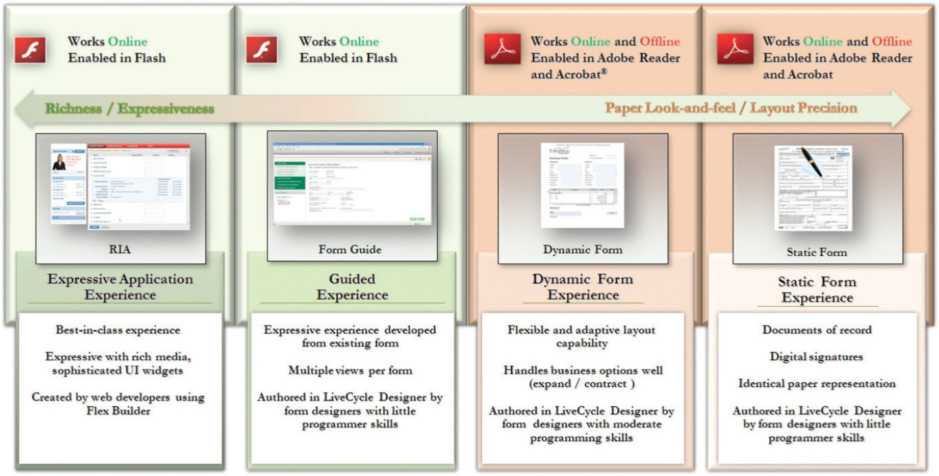
| Name | Data Type | Purpose | Input |
|---|---|---|---|
| xmlDataIn | xml | Process | ✓ |
| objTaskResultCollection | Task Result Collection | Process | |
| objHRReview | Task Result | Process | |

The LiveCycle Workbench ES2 Process Design view simplifies business process modeling with concepts such as swim lanes.

LiveCycle Workbench ES2 unifies authoring tasks for forms, guides, processes, document description XML (DDX), and data models. Integration with the repository simplifies management of artifacts, providing support for critical management activities, such as revision control, security, and auditing. The process design perspective gives business analysts the ability to create processes using standard Business Process Modeling Notation (BPMN) constructs (for example, swim lanes and pools). The form design perspective embeds the LiveCycle Designer ES2 functionality for the graphical layout of XML form templates that can be rendered as PDF, SWF, or HTML by LiveCycle Forms ES2. Forms are easily integrated with back-end data sources and destinations to build applications that automate common tasks, such as data capture. Users with existing XForms can leverage that investment using facilities to import XForms, convert to an XFA-based template, and render in HTML, PDF, and SWF formats. The data model perspective allows developers to write applications at a higher level, thus reducing development time and simplifying data integration. The guide builder perspective allows designers to author intuitive Flash-based data capture interfaces. The document builder perspective allows document authors to visually construct a DDX command files that will dynamically manipulate PDF documents. Based on Eclipse, LiveCycle Workbench ES2 minimizes up-front training, costs, and risks, and supports the Eclipse plug-in environment.

LiveCycle ES2 enables enterprises to deploy RIAs and guides. These are highly interactive, expressive interfaces for web and desktop applications that can run in a browser using Adobe Flash Player software or on the desktop with Adobe AIR. Engaging applications built with Flash Builder 4.0 and the underlying Flex framework can reach more users, improve productivity and satisfaction, and generate increased profits. These achievements are possible because LiveCycle ES2 integrates easily with Flash, leveraging the Flex remoting functionality exposed by LiveCycle ES2 services. The LiveCycle Service Discovery Plug-in for Flash Builder 4.0 allows developers to quickly build Flex and Air applications that invoke a LiveCycle service through its remoting endpoints. This allows these applications to easily use LiveCycle to process data, create documents and kick off processes.

The Flex framework provides more than 100 interface components. Flex user interfaces are described in industry standard MXML. Since Flex support is fully integrated with LiveCycle ES2, developers can take advantage of platform services to manage and deploy Flex components. Form designers can also utilize guides as a data capture interface to help users step through a task in a logical sequence. Guides are wizard-like applications that help people navigate through the data capture process. Guides can dynamically change and adapt based on input data to help ensure that only relevant and accurate data is captured. And with an intuitive WYSIWYG guide builder that leverages a common data model, designers2 can create guided experiences at a higher level, reducing the amount of code and simplifying data integration.


Works Online
Enabled in Flash
Works Online
Enabled in Flash
Works Online and Offline
Enabled in Adobe Reader and Acrobat®
Works Online and Offline
Enabled in Adobe Reader and Acrobat
Richness / Expressiveness
Paper Look-and-feel / Layout Precision
RIA
Form Guide
Dynamic Form
Static Form
Expressive Application Experience
Guided Experience
Dynamic Form Experience
Static Form Experience
Best-in-class experience
Expressive with rich media, sophisticated UI widgets
Created by web developers using Flex Builder
Expressive experience developed from existing form
Multiple views per form
Authored in LiveCycle Designer by form designers with little programmer skills
Flexible and adaptive layout capability
Handles business options well (expand / contract )
Authored in LiveCycle Designer by form designers with moderate programming skills
Documents of record
Digital signatures
Identical paper representation
Authored in LiveCycle Designer by form designers with little programmer skills

LiveCycle ES2 allows appropriate interfaces to be created with a common forms and data model to improve flexibility and engagement.

While the request/response communication model is sufficient for website browsing, many applications require optimized high-performance data transfer as well as additional modes of interaction, including publish/subscribe messaging and the ability to push data or alerts from the server to the client. The LiveCycle Data Services ES2 solution module enables high-volume data exchange guaranteed delivery and synchronization between AIR, Flex, or Ajax based application interfaces and back-end systems.

With the introduction of the new Adobe application modeling technology, developers can write applications at a higher level, reducing development time and simplifying data integration. LiveCycle Data Services ES2 is deployed as a standard J2EE web application, so Java™ developers can leverage their existing business logic by enabling applications to publish and subscribe to JMS topics and queues and to integrate with existing EJB components. Backed by a powerful data management API and connectors, the module simplifies client-server data communication challenges from multiple sources, including collaborative content shared across clients in real time.

Adobe application modeling technology is similar to other data modeling languages in that it supports a mechanism to define data entities and relationships. However, this capability is not adequate when it comes to building today's data-centric applications. Adobe application modeling technology fills the gap by adding support for services and data behaviors. Specifically, the model becomes a representation of the application, instead of a representation of your data. This enables the various components of your application to share data and business logic.

### Deploying

LiveCycle ES2 application commissioning tasks related to testing, staging, and deploying to production servers are simplified through the use of LiveCycle Archive (LCA) files that automatically package application-related resources in a single archive for transfer between individuals and systems. Administrators can stage or deploy applications to production servers using LiveCycle administrative tools. No development infrastructure is required on the servers to which applications are deployed.

### Running applications

The LiveCycle ES2 architecture embraces SOA principles that enhance an organization's ability to organize and use distributed capabilities. As described in this paper, LiveCycle ES2 provides a framework in which business analysts, application architects, form designers and developers can match business needs with capabilities by exposing back-end processes as services that are readily assembled into targeted solutions, greatly improving an organization's agility to meet changing business needs.

LiveCycle ES2 exposes business functionality as discrete services with clearly defined interfaces that allow them to be repurposed and used as required by other applications. For example, LiveCycle Rights Management ES2 provides the same robust, document-level access control to all applications requiring this functionality. Since exposed services often operate across domains of ownership, organizations must be able to declare policies that describe a service in technical and business terms, encompassing security, governance, and related functions. LiveCycle ES2 services are deployed with XML descriptions that qualify their use.

### LiveCycle Service Container

The LiveCycle Service Container is a single, scalable, and unified runtime environment based on SOA principles, in which all standard and custom LiveCycle services execute. The Service Container functionality includes orchestration of services into synchronous and asynchronous processes, monitoring, auditing, and coordinated security. Developers and administrators also benefit from simplified and fine-grained development, deployment, securing, and maintenance of services. When an asynchronous client request has been authenticated, it is passed to the Job Manager, which invokes the service.

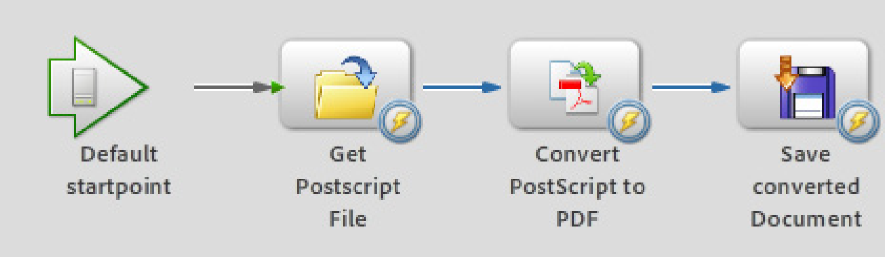
Several key attributes of the Service Container are described below:

- **Component model:** The Service Container employs an extensible component model in which loosely coupled services can interact to provide compelling customer engagement experiences. LiveCycle Foundation provides services that enable the LiveCycle ES2 platform to integrate with common enterprise infrastructure, such as directory servers over LDAP, web services, JDBC, FTP, and file systems. LiveCycle ES2 also encompasses standard solution services such as LiveCycle Reader Extensions ES2 to enable enhanced features in Adobe Reader, and

LiveCycle Data Services ES2 to integrate customer engagement applications with LiveCycle services and back-end systems.

- **Orchestration:** A key aspect of the LiveCycle ES2 runtime environment is a set of facilities that orchestrate processes to complete complex tasks. Organizations can orchestrate multiple processes together into a new service, provide end-user interaction with LiveCycle ES2 services as part of long-lived business processes, and integrate services with third-party products and services.

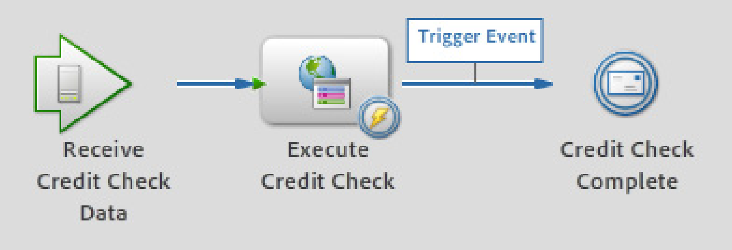
By orchestrating existing document services, solutions can be easily assembled and tailored to specific requirements. For example, consider a requirement to retrieve a PostScript® file from disk, convert it to a PDF file, and save it back to another disk location. Developers can easily meet this requirement using LiveCycle Workbench ES2 to combine three existing services into a single, orchestrated service called ConvertPS. This service uses a single invoke operation and takes full advantage of the invocation methods available to LiveCycle ES2.



ConvertPS orchestrated service

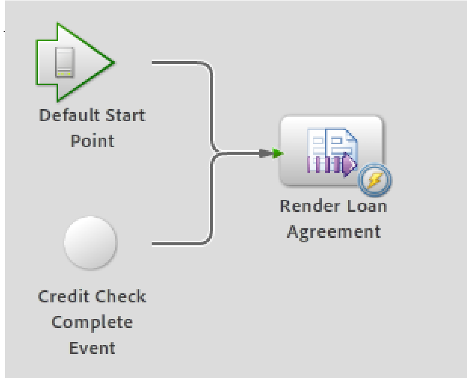
- **Event framework:** LiveCycle ES2 provides a framework through which solution developers can raise and receive customized events to facilitate communication between running processes. Events can be used to accomplish tasks such as initiating, terminating, or causing a state change in a process when a dependent operation in a related process completes. Information about an event is stored in an XML variable available to the event receiver. Events can be filtered to determine which event filters are triggered. LiveCycle ES2 supports asynchronous, exception, and timed events.

A representative process utilizing the event framework is a loan application in which a CreditCheckComplete asynchronous event is thrown after the credit check completes. This process is initiated when the customer submits a loan application.



The loan application process uses the event framework to asynchronously signal completion of a credit check.

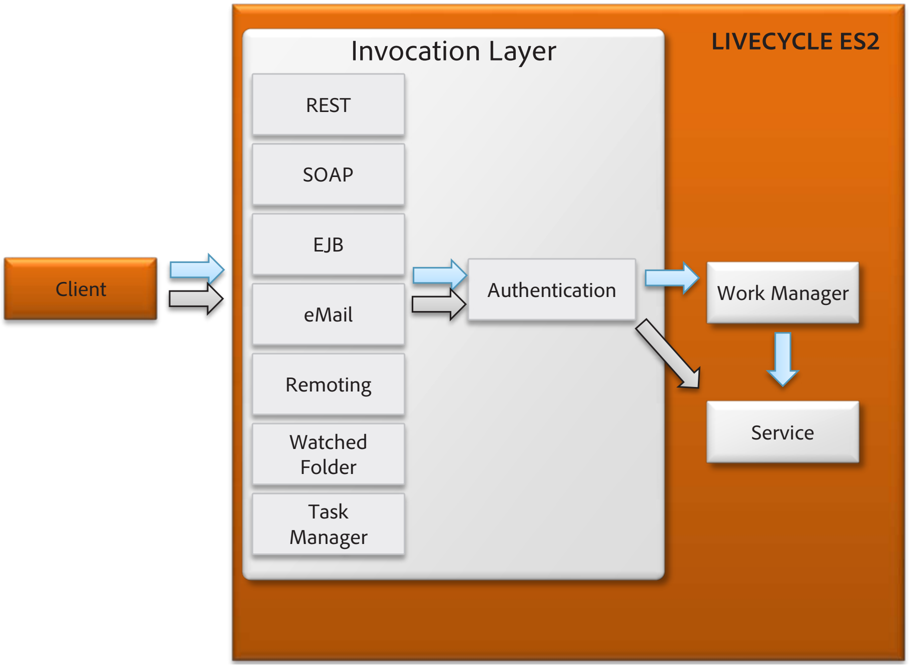
The CreditCheckComplete event receiver functions as the starting point for the subsequent ApproveApplicant process.



Upon completion of the credit check, additional processes may be invoked, automating additional business activities.

### Invocation layer

The LiveCycle ES2 invocation layer allows people and external applications to initiate processes via a wide range of mechanisms. The invocation layer intercepts invocation requests, performs security, auditing, logging, and transaction management functions and ultimately invokes a target service. Any client, such as .NET, PHP, or e-mail, can start a service. This unified invocation framework reduces complexity and improves interoperability.



Flow of activity when a client makes an asynchronous request to a LiveCycle ES2 service is represented by the blue arrows. Synchronous communication is represented by the gray arrows.

LiveCycle ES2 provides several integration methods for invoking a service:

- **REST:** Processes created in LiveCycle ES2 can be invoked through Representational State Transfer (REST) requests. REST requests are sent from HTML pages-that is, LiveCycle ES2 services can be accessed directly from a web page using a REST request. For example, any web browser can be used to invoke a LiveCycle ES2 process and load a rendered PDF document with data that was sent in an HTTP POST request.
- **SOAP:** Service invocation via web services is Web Services Basic Profile 1.1 compliant. It supports Simple Object Access Protocol (SOAP) with Base64, MIME, DIME, HTTP, MTOM, and Swa Ref attachments, the WS Security 1.0 standard, dynamic generation of SOAP endpoints and WSDL. Web service invocation allows both synchronous (short-lived) and asynchronous (long-lived) service requests.
- **EJB:** Service invocation via Java utilizes Remote Method Invocation (RMI) and utilizes a strongly typed API. Java-based invocation methods are highly optimized for document throughput and provide excellent performance when submitting a large document to a service.
- **E-mail:** A service can be invoked when a specified e-mail account receives an e-mail. This invocation method supports IMAP and POP3 using both basic authentication and SSL. Designers can define patterns for e-mail attachments and map data from e-mail headers, body, and attachments. For example, a PDF file e-mailed to a specified account could invoke an “apply policy” operation and return the PDF file to the sending e-mail address.
- **Watched folders:** Services can be invoked by writing files to network folders that LiveCycle ES2 has been configured to watch. The folders are scanned using configurable filename patterns such as .pdf and a desired processing action taken. Processed output files can be written back to a designated folder for consumption by external applications or users.
- **Remoting:** Flex clients can call LiveCycle ES2 services via the Flex RemoteObject tag. This method provides ActionScript to LiveCycle ES2 bridging, supports synchronous and asynchronous invocations, and provides dynamic endpoint creation, saving substantial development resources.
- **Task manager:** This integration method is only available in LiveCycle Process Management ES2. By exposing a service as a task in LiveCycle Workspace ES2, users with appropriate access rights can manually invoke that task from the Workspace Start Processes list. For example, an employee returning from a business trip could start an expense submission process.

After an asynchronous request is received via one of these invocation methods, the message receiver authenticates the request and creates a normalized invocation request. It looks up the service and operation name and passes this information to the router.

### LiveCycle Foundation

LiveCycle Foundation provides underlying functionality common to all applications, including routing and management of requests from clients. When an asynchronous request’s service and operation details are retrieved, the router calls the invocation manager. A job ID is then returned to the client through the receiver.

### LiveCycle Work Manager

Operations performed by services in LiveCycle ES2 can be either short-lived or long-lived. Short-lived operations complete synchronously on the same thread from which they were invoked. Familiar to most programming languages, these operations wait for a response before continuing.

However, many tasks performed in enterprise processes cannot be expected to complete synchronously. These long-lived operations often span systems or even extend beyond the organization, such as when a client must complete and submit a loan application form as part of a larger solution that integrates multiple automated and human tasks. Such operations must continue while awaiting a response. Long-lived operations perform their underlying work asynchronously, permitting resources to be otherwise engaged while awaiting completion. Unlike a short-lived operation, the LiveCycle Work Manager does not consider a long-lived operation complete once it is invoked. An external trigger, such as a system requesting another operation on the same service or a user submitting a form, must occur to complete the operation. The LiveCycle Work Manager utilizes industry-standard JSR-237 Work Manager for Application Servers to receive status information. LiveCycle ES2 provides a ready-to-use framework for long-lived operations, including tracking and recovery.

### Registry and repository

The LiveCycle ES2 repository provides advanced facilities for application developers that greatly enhance the accessibility and management of assets. The repository seamlessly spans single or multiple clustered J2EE systems, simplifying scalable and failover deployments. Versioning of deployment assets helps structure and control these activities at runtime. Components and services in the central repository are inventoried in respective registries that facilitate browsing, lookup, starting, and stopping components and services, and deploying and undeploying services.

Adobe LiveCycle Connectors for ECM provide simple integration between LiveCycle ES2 and ECM systems such as EMC Documentum, IBM FileNet, IBM Content Manager, and Miscrosoft SharePoint.

When a service is invoked, the registry and repository are utilized to access runtime assets required to complete the request. Clients poll the status of their asynchronous requests and when the job is marked as complete, they call the service manager to retrieve the result.

### Integrating applications in the enterprise

LiveCycle Foundation provides a comprehensive range of methods for integrating, both as a server and as a client of other processes.

In addition to the invocation methods described above, LiveCycle ES2 supports several methods for requesting the services of other processes in the enterprise and returning the result to an application. These methods provide flexibility to leverage existing resources.

- **Web services:** A service can also be implemented as a proxy to an existing web service. The proxy definition in the component.xml file contains the information necessary to define the operations and parameters to be exposed as part of the proxy service.
- **Custom components:** Developers can add their own services by creating stateless plain old Java objects (POJOs) containing one or more operations. Java developers can often reuse existing code with minimal effort. The POJO, a component.xml file describing it, and any dependencies, such as libraries, are combined into a single JAR file. The document service is then deployed to the LiveCycle Service Container, using familiar tools and steps as in any other deployment.
- **Connectors for ECM:** Connectors provide content repository services such as check in/out, lock, and content metadata access. The connector service components receive and output document and other content objects through the LiveCycle process, connecting ECM to customer applications.
- **JMS:** LiveCycle ES2 supports the JMS API for communicating with other Java-based applications in a distributed environment.
- **Service provider interface:** LiveCycle ES2 includes a service provider interface (SPI) to standard external authentication methods (Lightweight Directory Access Protocol (LDAP) servers, Microsoft Active Directory, and custom authentication and authorization methods, for example).
- **Foundation services:** LiveCycle ES2 supports a number of out-of-box components and libraries that provide core functionality for integration with common IT infrastructures. Support includes integration to the user directory through LDAP, authentication, ECM, JMS, SOAP, e-mail, and other back-end systems, applications, and resources.

- **LiveCycle Data Services ES2:** This solution module provides real-time messaging, guaranteed messaging, quality of service, and data management services for high-volume data exchange and synchronization between Flex or Ajax-based application interfaces and back-end systems.

### Administering applications

The LiveCycle Administration Console is a web-accessible tool that system administrators use to perform a wide range of tasks, such as adjusting port number and logging options, and managing the deployment and configuration of LiveCycle ES2 applications. Administrators also use the LiveCycle Administration Console to create and manage solution components service endpoints and to deploy applications. It also provides a common authentication management service used throughout the platform and across all services, easing tasks such as defining and configuring users and groups and integration with LDAPs. Administrators also have access to JMX-compatible service monitoring via the LiveCycle Administration Console.

Administrators can also export, import and manage LCA files that contain an application's process definitions and related resources such as forms, images, and XML files. LCAs simplify the export and transfer of applications between systems, and can be imported into LiveCycle ES2 without stopping the system. This mechanism, which also supports scripted deployment, eases the workload for organizations wanting to stage solutions from development through testing and deployment, and simplifies the sharing and distribution of applications with partners and customers.

The Business Activity Monitoring Workbench allows system administrators to quickly set up business connectivity monitoring by configuring event and contextual data sources and accessing multiple, concurrent data streams. System administrators can also create business rules and dashboard objects, as well as views and cubes in the BAM Workbench. Administrators can define criteria ranging from simple to complex in order to trigger alerts.

### Summary

LiveCycle ES2 is a scalable, unified platform that blends data capture, business process management, document security, and document generation to help enterprises create and deliver rich and engaging applications that reduce paperwork, accelerate decision-making, and ensure regulatory compliance. LiveCycle ES2 provides the tools to develop and deliver applications in a single, J2EE-compatible and standards-based environment that eases integration of LiveCycle ES2 applications with existing IT assets.

Organizations can use LiveCycle ES2 to solve diverse problems ranging from data capture to complex business transformations. For example, government agencies support processes ranging from electronic tax submissions to full eGrant application management. Financial services solutions include loan application processing and account opening that help institutions distinguish themselves in a highly commoditized market.

LiveCycle ES2 eases development through an integrated toolset that enables business analysts, developers, form designers, and others to work collaboratively with shared, reusable resources. Central to LiveCycle ES2 is a single runtime environment that provides invocation, event management, process management, and other essential functions to foundation and solution modules. Integration with users and external processes is facilitated through common invocation and request methods such as Flex remoting and web services.

The service-oriented design of LiveCycle ES2 greatly simplifies the assembly, deployment, and maintenance of customer engagement applications, combining rich and effective user experiences with back-end processes exposed as services in a consistent way. By assembling customer engagement applications from loosely coupled services in the LiveCycle ES2 platform, organizations can reduce development, deployment, and maintenance complexity, align IT systems with business priorities, reduce costs, and improve agility and return on investment.

**Adobe Systems Incorporated**
345 Park Avenue
San Jose, CA 95110-2704
USA
*www.adobe.com*

95010880 11/09