In today’s environment, every business action is tied to a business process, set of records, social interaction, exchange of information or discovery activities. The productivity of an enterprise is an affectation of its capacity to consume, interpret and take action on information. A business action or strategic goal can be dissolved down to a work breakdown structure containing discreet tasks for individual workers or roles within the organization. Tasks are a good allegory for Apps. The can be simple, designed to get a travel approval or they can be complex, designed for insurance claims management. Here we investigate the concepts and employments of Enterprise Application Development and explore how integrated Enterprise Information Management platforms and user-centric apps, built with efficiency and experience in mind can change the way we look at the IT investments we’ve made over the past years and those to come.
# Table of Contents

Introduction ................................................... 3

The Definition of Enterprise Apps ......................... 7
  *Eight Key Takeaways On Defining Enterprise Apps* ............ 8
  *Developing an AppWorks App or Service* ...................... 9
  *The AppWorks Stack* .................................... 13
  *AppWorks Architecture* .................................. 17

App Modernization ........................................... 17

The AppWorks Developer ..................................... 19

The AppWorks Gateway ....................................... 20

The OpenText Common RESTful API ......................... 22
  *API Case Considerations* .................................. 22
  *URL Structure* ........................................... 23
  *Verbs* .................................................. 26
  *Response Formats* ....................................... 29

AppWorks and The Future .................................... 30

Contact OpenText ............................................ 31
Introduction

This paper investigates the concepts and employments of Enterprise Application Development Platforms. Through it, we will explore how integrated Enterprise Information Management platforms and user-centric apps, built with efficiency and experience in mind can change the way we look at the IT investments we’ve made over the past years and those to come.

An ideal Enterprise Apps platform would be designed to hybridize the roles of technical and business transformation as seen in some of world’s most successful organizations, which will help us to ‘infuse IT into conversations about new business ideas from the start’.

To understand the importance of a sound Apps platform and its relevance to business and technical transformation, we need to understand what role existing IT investments play in business transformation, which can be defined as the act of radically changing the series of actions required to meet a specific business goal. This includes the processes, procedures and policies of an organization as well as the strategic targets, content management and reuse ambitions, compliance and customer experience objectives it pursues.

In today’s environment, every business action is tied to a business process, set of records, social interaction, exchange of information or discovery activities. In other words; the productivity of a company is an affectation of its capacity to consume, interpret and take action on information. A business action or strategic goal can be dissolved down to a work breakdown structure (WBS) containing discreet tasks for individual workers or roles within the organization. These tasks could range from insurance claims management to media publishing and creative review, other examples could be a procurement procedure or records compliance activity within an organization, a retail kiosk capability, a service bookings and customer service web experience or even a field inspection app. Tasks are a good allegory for Apps. Apps can be simple, designed to get a travel approval or expense procedure. Apps can be complex, designed for commercial lending approvals or insurance claims management.

To handle these tasks (or apps) today, companies implement procedures which consist of electronic and automated business processes, paper or doctrine based processes, enterprise content management systems, business process management systems, media management, web content management and so forth, each targeted at solving a specific set or subset of tasks associated with the operations of the company and the attainment of its goals.

The drawback to this approach is that users constantly need to learn the capabilities of complex systems while interacting with multiple departments through multiple mediums, all of which comes at the cost of confusion, isolation and lost productivity. The concept of an Enterprise Apps platform is to provide developers with the capacity to integrate any number of back end systems to develop a task-solving app, which users can interact with and is designed to execute the task in question with the utmost efficiency, while leveraging the existing infrastructure and systems of the organization. As an example; a procurement procedure, which requires the submission of a statement of work reviewed by the legal department, a purchase order, the vendor’s contact information and a written description of the work being done as well as the department being billed. This process requires the contract to be emailed to legal, then back to the initiator, then to procurement where they add an executed, scanned copy to the legal department’s contract shared drive. The initiator must then go to an ERP system to fill out a purchase order requisition before emailing a word document with the project and vendor details to a group mailbox.
This process is fraught with possible productivity leakage, poor compliance and confusion amongst staff. The simple solution is to create an Enterprise App leveraging BPM, ECM and iX API’s to make a quick and user-focused HTML5 app, which accommodates this task in its entirety, while still remaining compliant with the aforementioned business procedures within the company. In fact, all of the involved departments could use the same app, which was written once and is now available on the web, on smartphones and tablet devices and also on the user’s desktop. An Enterprise Application Platform takes care of the API orchestration and integration necessary to ensure all of the processes are executed and the artifacts stored in the requisite corporate repositories. To the end user, it’s a simple procurement app, where all that’s needed is displayed to the user in an intuitive and simple way as an App. An example of some the benefits asserted by using an Enterprise App to resolve this task within the organization:

- A quick time to production, thanks to HTML5/CSS and JavaScript client development languages.
- The ability to use existing on premises or cloud application environments to build NEW multi-platform, easy to manage and up to date apps that help to transform your business.
- A simple and secure path to building an Enterprise Application using a common RESTful API to access your Enterprise Information systems.
- The ability to install, update and manage Enterprise Apps centrally and without user intervention.
- Services designed to further help developers build secure, managed Enterprise apps for the web, mobile devices and desktops such as:
  - Authentication
  - Push Notifications
  - Service Management
  - Trusted Server Management
  - Usage Reporting
  - Application Management
  - Device Management
AppWorks: The Power of EIM Applications

The status quo for years in Enterprise computing and Information Technology was to avoid customization at all costs and to force the employee and user bases to work within the interfaces of existing Enterprise systems while switching from place to place within business processes that were loosely enforced. As such the number of deployed systems has become significant and the cumulative investment commensurate. These systems have been effective at accomplishing the objectives for which they were deployed and global productivity has correspondingly increased as a result of the information and industrial ages.

There is a shift occurring, however, as organizations become more complex in order to adapt to the global economy. Individuals become more specialized and we see in business a microcosm of the organic solidarity characteristics which have developed as an affectation of our complex social interdependencies and high degree of field or task specialization, which helps to drive productivity at the macro level of an organization. For example, an expert Records Manager will ensure that management and compliance are achieved for regulatory purposes deftly and for an organization of hundreds. However, if this task were to be delegated out to each individual within that organization as a fractional allotment of their workload, the cumulative effect would be much greater on the organization’s efficiency and the cost to achieve compliance much higher.

This happens because the hundreds of other employees are not specialized in the practice of records management to the degree of the aforementioned Records Manager nor are they as familiar with the systems utilized to accomplish this task. As such, the time it would take each employee to complete their fractional allotment of the overall compliance requirements would add to up to much more expense for the company than the salary of the specialized Records Manager. To draw a simple analogy: In a small village a full time, specialized fisherman will more efficiently feed the populace than having each village member spend a portion of their time catching a single fish each day for supper. This specialization also means that the other members of the community may specialize themselves to accomplish other tasks with as much effectiveness.

![Gross World Product](image-url)
In every organization we see the effects of organic solidarity in the very DNA of how we arrange our departments. We have legal specialists, IT specialists, sales specialists, marketing and engineers. Each with their own skillsets and each with their own unique cognitive filters established through years of specialization and operation within a given field. This practice of organic solidarity has not only allowed us to reach unprecedented levels of productivity, it has also enabled both our businesses and society to attain the utmost in capability, execution, safety and retention of knowledge.

The limitations surrounding a similar ‘App Specialization’, perhaps inorganic solidarity, until now have been technological, preventing us from enabling information workers with the same efficiencies of specialization. We’ve been awaiting the right tool for the right job. With a sound Enterprise Apps Platform, you can provide all of the conveniences needed to make your workforce a force of nature. Effective, efficient, compliant, secure and current solutions to complex tasks, helping you to reach your goals on a large scale, that’s what Enterprise Apps platforms offer. A medium for Enterprise transform their business, reduce their time to competitive advantage, while leveraging and modernizing existing IT and EIM investments and infrastructure.

The time has come to turn the tables on the systems, processes, content and information that run our organizations and provide a platform, which bends to the will of the people and breaks down the development, integration and user experience barriers that are the consequences of such progressed specialization and isolation. A platform to build, integrate and experience the lifeblood of an organization in a way that makes sense to your village and its specialized workforce. That platform is AppWorks.
The Definition of Enterprise Apps

Enterprise Applications are very different than their consumer counterparts. Enterprise Apps aren’t built exclusively for mobile devices, they run on web browsers, desktop computers, laptops, tablets and smartphones at once. Most information workers still spend their day in front of a desktop or laptop computer in a web browser. It can’t be underestimated how important it is to provide apps that work for the Enterprise, and this means bringing apps to the user, where they work.

What really differentiates Enterprise Apps is that they are secure. They’re created by professional developers using the capabilities of in place Enterprise Grade systems such as Enterprise Content Management (ECM), Business Process Management (BPM), Information Exchange (IX), Discovery or even ERP, CRM or other systems. They require authentication against your directory server and adhere to the security, compliance and procedural policies of the organization.

OpenText’s Enterprise Apps Platform, known as AppWorks provides a standardized RESTful API, common authentication, notification, audience management and other services to make developing apps against your EIM systems fast, simple and inexpensive.

The AppWorks platform allows developers to make apps in HTML5/CSS and JavaScript in the IDE of their choice, using the tools and frameworks or bootstraps of their preference and then deploy that app at once to users on an array of device profiles such as web, mobile and in the future, desktop.

When developing apps, HTML templates can be used or reused, components swapped in and out, experience and appearance changed as simply as on any web page. These apps can then be deployed out to groups of users managed in a directory server to ensure Enterprise security and IT user policies are enforced.

Enterprise Apps are different than consumer apps because they are designed to solve business problems, leveraging the systems and processes already in place. They are quick and cost effective to create, tune and update, making them the clay through which changes to the business may be molded without the requirement for substantial re-architecting of corporate systems. Apps contain the experience and diffuse the complexity of data management to the platform and in-place back end systems in a compliant and simple fashion. This is accomplished through both the OpenText Common API furnished by AppWorks and through the capacity to install and manage custom JAVA services for Apache Tomcat directly through the Gateway’s administrative interface. These services can orchestrate API interactions, offload complex actions from client devices or access the capabilities of your EIM platform.
Eight Key Takeaways on Defining Enterprise Apps:

1. Enterprise Apps are inexpensive and easy to build HTML5/CSS and JavaScript apps, which run on web browsers, Android and Apple mobile devices and in the future Windows and Mac desktop. They need only be developed once (not once for each platform).

2. Enterprise Apps are built using the power of your OTi and EIM solutions, maximizing your investments, and accessible for developers through a common API delivered through AppWorks.

3. Enterprise Apps are secure, managed, audited through a centralized administrative interface, which provides the ability to:
   - Report on App, Device and User activity
   - Wipe Devices Remotely (BYOD Safe, Enterprise Wipe Only)
   - Deploy and update Applications without the user needing to take any action.

4. Enterprise Apps are POWERED by your Enterprise Solutions. You can quickly and cost effectively create secure, integrated applications that solve problems fast, require little or no training and will update / read data from your in place enterprise environment.

5. Enterprise Apps drive productivity, compliance, accessibility, modernization efforts and other transformation initiatives.

6. Enterprise Apps can leverage the native capabilities of the device they are being run on, such as a tablet camera, accelerometer, desktop filesystem, drag & drop, webcam and other capabilities.
   - For mobile devices, Apache Cordova is provided with the AppWorks managed clients.
   - For Web and Desktop, OpenText proprietary integrations are made available in the pattern of Cordova.
   - Other technologies of the developer’s preference may also be used and included in apps to accomplish this capability for specialized clients, such as a retail or manufacturing kiosk, etc.

7. Enterprise Apps can be very small things, such as an approval request or site inspection photo attachment and submission tool or can be very large things such as a company’s secure social network or product catalog. Even the AppWorks Developer network is itself, an AppWorks App.

8. Enterprise Apps can be built, deployed, rebuilt from templates all quickly and cost effectively. They’re just HTML5 after all. This allows developers the flexibility they need to align to the changing needs of the business without having to worry about API backwards compatibility, interface retooling and expensive native client API integrations.
Developing an AppWorks App or Service

AppWorks Apps are composed using HTML5 markup and Cascading Style Sheets (CSS). JavaScript provides the client-side heavy-lifting and API integration capabilities.

Let’s begin with HTML, which is used for structuring and presenting content on the web. AppWorks allows developers to use HTML, which is very efficient for authoring user experiences and content layout to create the experience for their users. This is contrary to the way Enterprise systems typically manifest their experience layer; where the user interface is immutable or only selectively customizable, and in some cases totally inflexible. The old paradigm of working within multiple native Enterprise system interfaces also means huge amounts of productivity are lost due to complexity and forcing users to solve a problem within the framework of a generalized and complex Enterprise system.

Allowing developers to break free from simply extending and customizing existing platforms means they can start to build applications and solutions that suit the needs of the business. It changes the nature of the discussion from a dialogue of compromise and retrofit to one of business transformation and integration.
For the interface layer, HTML and CSS alone is not enough, so AppWorks supports the use of JavaScript within its clients. This means that AppWorks developers have a rich interpreted language available to them for development both within the clients and server-side services layer (if so desired) of the AppWorks Gateway. This JavaScript and HTML5 capability means that AppWorks developers have what they need to provide a compelling user experience, which is capable of interacting with a common RESTful API to move data to and from their applications.

In the case of AppWorks, the communication between the client and Gateway usually means leveraging the common RESTful API provided through the Gateway itself. This RESTful API gives developers a single point of interaction and single URI and authentication layer to move information to and from the end users working within the compelling and dynamic HTML5/ CSS/JavaScript user experience.
At the Gateway itself, Developers who don’t wish to leverage the OpenText RESTful common API may extend the services and API’s of 3rd party, proprietary or even OpenText solutions to fit the needs of their applications. This process of using a Service on the AppWorks gateway, which can be developed in Java / JavaScript can coordinate and marshal API’s, offload work to the server from the client, transform data or interactions or extend capabilities of other repositories for utilization in AppWorks Apps. This process is known as API Orchestration. AppWorks Services drop-ins are installable modules via the AppWorks administration interface. These packages are simply Tomcat WAR files with an AppWorks properties file and appropriate icon. This allows the developer a close to unlimited capability in terms of orchestration and service development capabilities.

The Common API is a web service, and as such is designed to leverage secure socks layer (SSL) and other typical over-the-wire encryption technologies. This ensures data transmitted to and from the gateway is secure.

Lastly in the layer cake, between the AppWorks Gateway and your existing Environment upon which Apps are being developed, there are private API’s. Private API’s are the native API’s of the solutions being integrated and can either be called directly by AppWorks Services or may be proxied through the gateway itself directly to the application being developed.

AppWorks is designed to be flexible, allowing business and developers to make the best secure, managed enterprise applications available to their user base with the best possible efficiency and experience, designed to drive adoption and continued utilization.

It should be noted that AppWorks further enables developers to expedite the speed to market by offering several centralized services out of the box, such as common EIM Suite Authentication, Audience Management, Feeds, Notifications and others. This gives developers the tools they need to make integrated apps with smooth experiences quickly and simply without having to rebuild complex services capabilities for each application.

The diagram aside this text illustrates some of the services built into the Gateway, which is itself a lightweight Tomcat Application.

Lastly on the definition of an installable and deployable application is the application payload itself. Again we’ve kept it standard and simple with AppWorks. Our application packages are created by simply zipping up your app developed using HTML5/CSS and JavaScript. You take your web application’s folders, zip them up and add a properties file and icon to the package before zipping once more, that’s it. It’s ready for deployment. This structure also allows developers to include static content required by applications to minimize the time to action experienced by end users. Once downloaded to the user’s managed client, the content is available from the managed enterprise cache that AppWorks maintains.
Above we see an illustration of the ‘app payload’, containing folders with style sheets, images, JavaScript assets, and other content. The green box surrounding it is the first zip file. You have simply taken the folder and file structure of your web app and zipped the entire thing up. Index.html is the entry point for AppWorks applications; from there you navigate within your application using the experience you have designed. An AppWorks app can provide functionality as wide or as narrow as the business needs dictate.

Once you have zipped up your HTML5 app, you create and provide an icon for your application and a properties file to tell the Gateway whatever your app needs to know on installation. Once complete, you zip the entire payload up and it’s ready for installation on an AppWorks Gateway. The final zip container is represented by the orange box in the figure directly above.

As you can see, it’s incredibly easy to create Apps and integrations using AppWorks. References for API’s, sample code, tutorials and other information to support developers is available in OpenText’s new AppWorks Developer Network at developer.opentext.com
The AppWorks Stack

The platform stack utilized for a production implementation of AppWorks has been selected to afford maximum scalability, lowest cost and the most robust configuration possible. The stack described hereunder is the recommended deployment configuration and the one utilized by OpenText to power AppWorks Developer. Other web servers, databases and operating systems may, of course, be used depending on the considerations of the IT department or group deploying AppWorks.
Starting at the top of the stack, closest to the end user, AppWorks utilizes Nginx as a free, open-source high performance HTTP server and reverse proxy server. Nginx is designed to utilize a very low memory footprint and is used to power some 36% of the top 10,000 websites globally and ~37% of the top 1,000 websites thanks to its extreme performance and reliability, according to W3Techs.

After the web server in the AppWorks Stack comes the selection of the application server and optional high-throughput services platform. For AppWorks, Apache Tomcat is a must. Tomcat is used to host the AppWorks Gateway application itself, from there AppWorks provides the UI required for installation, management and updating of Apps and Services, which in themselves are zipped up web applications and Tomcat WAR packages with the properties and icon files required by the AppWorks Gateway, similar to the application payload described above. In keeping with the Nginx specifications, Tomcat is also a highly scalable, stable and free open-source product.

As a supplement to Tomcat, if the applications or integrations being created in AppWorks require some very intensive server side processes for API orchestration, computational functions, machine-to-machine interaction or other capabilities, an optional high-throughput software platform for server side scalable network applications may be used. OpenText recommends node.js for this capability due to its non-blocking, highly scalable architecture and JavaScript language basis. Node.js is also a free open source product available under the MIT license construct.

On the second last layer of the AppWorks Architecture sits the database. This maintains server configuration, application metadata and manifests and other information needed by the AppWorks Gateway itself. OpenText uses and recommends MySQL. MySQL is the world’s most popularly used open source relational database management system (RDBMS). MySQL is recommended for AppWorks because it is free, open-source, scalable, robust and trusted the world over. AppWorks will also work with MSSQL and Oracle databases to accommodate popular Enterprise Standards and already available licenses and infrastructure for Developers who utilize those products within their IT environments.

Lastly is the Operating System for AppWorks. OpenText uses and recommends Linux for this purpose thanks to its high reliability and presence as a standard for Enterprise deployments the world over. OpenText AppWorks Developer is powered and run by Linux, thus completing the recommended stack configuration for OpenText AppWorks from web server to Operating System.

It should be noted on the topic of the AppWorks Stack, that customers may utilize the components that work best for them and their environments. As an example, the Linux distribution utilized may vary from customer to customer, or some may prefer to use the Apache web server and have no requirement for node.js.

AppWorks is designed to scale from a Raspberry Pi all the way up to a massive and redundant enterprise server farm operating as a cluster. The key is flexibility and simplicity with AppWorks. Developers need the option to size the platform for their applications and IT needs the option to reduce costs and deliver a better quality of service for Business users – this is what AppWorks makes possible.
AppWorks has an architecture designed to make the development, deployment and management of Enterprise Apps and Integration Services quick, simple and effective. Within the web server powering AppWorks, three primary things are happening:

1. Markup Content and user experience is being sent to and from the AppWorks Managed Clients. This is the layer at which the user ‘experiences’ an AppWorks App. The Web server is in place to ensure the response to the user is fast and pleasing and ensures the rich content your App is built with is served up to your users regardless of which device they may be on: web browsers, mobile phones or tablets or even Desktop Applications in the future. This capability is labeled ‘Pages’ within the Web Server block in the architectural diagram above.

2. Files and content required by applications are transferred from the gateway to the Clients through the webserver. If your app furnished PDF files or contracts to the user as an example, they will pass through the Web Server attached to AppWorks.

3. The Proxy service afforded by the Web Server and managed through the AppWorks UI allows developers to integrate their applications by moving API calls, web content and other data through the AppWorks gateway. This eliminates the need for additional routes through the firewall and means that developers have one URL to hit for all content, all apps and all services. Depending on the traffic a particular app sees, this proxy may be under a lot of stress, which is why OpenText recommends a highly scalable, non-blocking solutions such as NGINX.
Under the web server layer, we see what is described in the center of the diagram with a blue label as ‘Services’. This is AppWorks itself. To reiterate the description in the ‘AppWorks Stack’ section of this document, node.js is an optional platform developers may select, but the Java component represents Tomcat and is a requirement for the AppWorks Gateway. The Tomcat Server hosts AppWorks itself, and within AppWorks are a selection of services geared at bridging the gap between creating compelling Enterprise Applications and the reality of updating, managing and deploying capabilities to thousands of end users who may or may not be technically savvy.

These centralized AppWorks services include things like Application Management. This is exposed through the AppWorks Web GUI and allows administrators or developers to install, test, deploy and configure applications for their audiences.

Service Management allows developers to install Tomcat services which orchestrate API calls, provide server side integrations or computational offloading and other capabilities.

Tenant, Device and Runtime Management allow administrators and developers to manage things like tenant-ship, connected and formerly connected devices to perform remote wipes, report on AppWorks Gateway utilization or to understand the metrics of your AppWorks user base.

The services afforded by the AppWorks Gateway are of course not exclusively focused at administration and management. They’re there to make life easier for developers. A comprehensive set of services intended to speed up integrated, secure and managed application and integration development are also present in the Gateway and include capabilities such as:

- Common Authentication
- API and Web Content Proxy
- Common RESTful API for OpenText EIM Products
- Notification and Call-Back Services for Apps Developers
- Common Feeds Service
- Audience Management (User Permissions by App)
- Many others...

Because AppWorks is designed as an Enterprise Development platform, all of the common services are built with the Enterprise Developer and infrastructure in mind. This means that the AppWorks Authentication service can be integrated with your company’s Active Directory or LDAP server to make deploying your apps and securing them in accordance with organizational policy very simple.
**App Modernization**

Organizations across the globe recognize a continuous drive towards the Cloud, Mobility and Open Source, which causes issues for many of the existing applications that are available to the business user. Applications need to move from the desktop and web to a variety of mobile devices (smartphones and tablets), with demands from business users that they offer almost identical functionality across multiple platforms.

Legacy applications are often unsuitable for smartphones and tablets, where security becomes an even bigger issue as organizations are worried about losing control of their data. At the same time, IT managers need the applications to be easy to deploy, manage and monitor.

According to Gartner, 78% of current enterprise application portfolios will experience some level of change in the coming two years — retire, replace or upgrade — while only 22% of the portfolio will remain unchanged. To accommodate the quickly changing business requirements, application modernization, rather than the more costly replacement of applications, is preferred. Gartner’s definition of application modernization is described as the efforts to maintain and extend the useful life of applications within an enterprise and the modification of the underlying infrastructure to achieve desired IT or business outcomes.

Making business critical information available to users wherever they are, while ensuring that access to data is secure and controlled, have become requirements to stay on top of daily business events. Rather than replacing expensive enterprise applications, companies are seeking to maximize their investments in existing technology by modernizing their applications instead of replacing them. The applications must be manageable from a central location, with application updates that do not require users to download and install them manually, and seamless integration and interoperability with the back-end services.

OpenText is the leader in EIM, a comprehensive set of best practices and technologies that addresses the needs of information workers by providing them with the right information during decision-making, analysis, procedure definition or process execution, wherever they are, and any time of the day. When executed properly, a sound EIM strategy results in significant productivity and efficiency gains, ensuring accountability and transparency, with an engaging customer experience that is available on all platforms.

In today’s market, there continues to be significant growth in the application marketplace – mainly powered by the ever-increasing popularity of smart phones and tablets, as well as the growing demand for modernized and specialized enterprise apps. This surge has created a huge demand for application development and management platforms, and application programming interfaces (APIs) that act as gateways between devices and services, while not compromising system and data security within an enterprise.

Application modernization can be achieved by creating new and focused applications on top of existing services. These new applications access the back-end services through well-defined and easy-to-use APIs provided by an Enterprise Application Development Platform and are typically designed using modern, common technologies, perfectly suited for light-weight web access and mobile devices. Such a platform can also help to deploy and manage the applications from a central location, while ensuring seamless upgrades.

In the case of AppWorks, the OpenText EIM API is made available using a pragmatic RESTful standard as articulated in the OpenText Common RESTful API section of this whitepaper.
Why do you need Application Modernization?
Over the years, organizations have invested heavily in their enterprise applications. These applications are often complex, while most users require access to only a subset of offered functionality. The user experience is less suitable for people who are on the move and require access to business critical information from wherever they are. Application modernization allows organizations to create apps that focus on particular tasks and embrace modern smartphones and tablets as well as the traditional web and desktop interfaces we’re used to, without a need to upgrade or replace the existing back-office systems. The rewards are substantial: lower costs, reduced business risk and efficiency and productivity gains that will produce value for the organization.

The Business Value of Application Modernization
Companies are looking to extend the life of their EIM solutions and protect their investments. At the same time, they need to address the demands of their employees, who wish to access business information anywhere and anytime. It is imperative that the solution is designed to meet the company’s essential needs: productivity, flexibility, security and manageability.

Reduce Risk by modernizing applications
The costs associated with replacing existing business critical systems are significant, but pales in comparison with the compliance risk and litigation costs. By modernizing applications with an Enterprise Application Development platform, companies are able to reuse the data that is compliant and securely stored in the back-end systems while creating tailor-made solutions that address the organization’s needs head-on.

Increase the value of your business information assets through productivity and efficiency gains
Modernized applications are tailored for use on any platform, with a vastly enhanced user experience and access to functionality that is task-focused, whether the applications are used on the web, desktop or mobile device. This approach significantly enhances productivity and efficiency.

Considering the high cost and potential risks of the replacement of business critical information systems, solutions that allow for the continued use of the existing services, combined with cross-platform availability and a modern and engaging user experience, can return value immediately.
The AppWorks Developer

AppWorks Developer is the developer network, which supports the AppWorks platform. In the Developer Network, coders have access to all they need to get started developing applications on top of the OpenText EIM stack. This includes, but is not limited to:

### API References
Access to the reference for the OpenText Common API, enabling developers to build apps and experiences on top of the OpenText EIM platform. This RESTful API is accompanied by services in the AppWorks Gateway, which further accelerate integrated application development, such as: authentication, audience management, feeds, notifications, favorites and others.

### Tutorials
How-to’s on topics ranging from the installation and configuration of the AppWorks Gateway on your Tomcat server through to building complex, integrated EIM applications for your employee, customer or user base.

### Downloads
AppWorks Gateway is available to any developer or company with 10 free development licenses to get started. The AppWorks Developer network is free to join and use as well!

### Code Samples
Copy and paste code snippets and samples for basic functions like obtaining authentication against the AppWorks Gateway and basic style sheets for your apps.

### Templates
HTML, App and Services templates to accelerate development of your enterprise EIM applications.

### Question & Answer Forums
Get answers fast from community voting and OpenText Engineers, who are constantly in the AppWorks Developer site making sure developers have the answers they need to make secure, managed and compelling EIM applications.

### Idea Ranking Capabilities
Crowd-sourced innovation, the AppWorks Developer site will have a social capability to rank and de-rank ideas within the community. This capability is for raising important issues and ideas to the forefront for community discussion and action by OpenText. We want the voices of our developers to be heard.

What’s special about AppWorks Developer is that it IS an AppWorks Enterprise App. The AppWorks platform is so versatile, our developer network was built as an App and powered by OpenText EIM Solutions.

This decision enabled OpenText to build our developer network once, and have it available on all web and mobile devices, but more importantly, it’s fully integrated with OpenText’s EIM systems, which means the full breadth of the AppWorks knowledge base and OpenText EIM infrastructure is available to help developer find what they need to deliver great enterprise apps.

The AppWorks Developer site is available to any developer at no cost, to access the material they need and download the AppWorks platform themselves. Our intention is to add value for customers by increasing the accessibility of OpenText development technologies and to provide opportunity to our partners and the broader development community to show off their innovation to OpenText’s customer base of Fortune 2000 and government and tens of thousands of other inventive organizations.
The AppWorks Gateway

The AppWorks Gateway offers benefits as a valuable add-on to existing OpenText EIM solutions as well as a stand-alone product in a new market.

For existing Open Text customers, the AppWorks Gateway provides a single point of integration for all of our systems. With a single unified application programming interface (API) for communications, customers can begin writing their own applications which leverage the capabilities of Open Text products such as Content Server, Tempo Social, Tempo Box, Easylink, Cordys BPM or others. The AppWorks Gateway also provides the ability to unify the browser-based user interfaces (UI) of their deployed Open Text products by hosting a common Open Text header so that all of our products can share common functionality such as authentication, profile management and notifications.

In the new market of application deployment and management, the AppWorks Gateway offers the ability to deploy applications using common web technologies to mobile devices in a controlled and secure manner. Two major advantages are the control AppWorks grants over deployed applications as well as the simplicity and speed of app development.

The enterprise application ecosystem and market place continue to see significant growth, mainly powered by the ever-increasing popularity of smartphones and tablets. This surge in mobility has created a huge demand for Application Development and Management Platforms and Application Programming Interfaces (API) that act as data gateways between devices and services, without compromising system and data security within the Enterprise.

AppWorks Apps are written using common Web technologies (HTML5, CSS3 and JavaScript), built and compiled On Premises (in the Cloud for future versions of The Application Gateway), and distributed through the AppWorks Gateway or commercial application stores to enterprise users, partners or customers.

Our customers and partners have been asking for OpenText to open up and standardize its development platform to enable them access to their EIM services and to build Applications on these deployments that run on desktops, the web and mobile devices through an easy-to-use Application Programming Interface (API).

To better serve our customers instead of providing a tool that creates just standalone mobile applications, we have focused our efforts on a solution to create, deploy and manage apps that connect to OpenText services from all platforms. This is the OpenText AppWorks Gateway.

The AppWorks Gateway provides more out-of-the-box enterprise capability than Web and native toolkits and addresses more of the full software development life cycle; from application design, development and integration to testing, deployment and management.
AppWorks IS...

- The platform used by Open Text and its customers and partners to deliver innovative new products and cross-platform enterprise apps.
- A unified platform and API for accessing selected current and future OpenText EIM services
- Brandable Client Runtimes for all major platforms (Web browsers and Mobile: Android and OS).
- Platform Agnostic: Write the app once (HTML5, CSS3 and JavaScript) get it everywhere with no duplicate development needed.
- Simple delivery and management mechanism for both standalone Apps and Apps developed for the deployed EIM platform.
- Runs on your premises leveraging all of your network security layers and existing infrastructure.
- Makes the deployment of new apps, management and updates, quick, easy, portable and effective without user intervention necessary.
- Allows developers to leverage their IDE of choice, be it Eclipse, Visual Studio or an HTML5/CSS creative authoring suite.
- Supports developer tools like Sencha Touch, Apache Cordova and other preferred and modern development utilities and libraries

Key features of OpenText AppWorks Gateway include:

- **Secure application management** – Complete control over which users have access to each application, the ability to enable and disable applications remotely and remote-wipe capability which gives administrators the power to remove applications and their data from user devices.
- **Write-Once application deployment** – Applications can be written using standard web technologies (HTML/CSS/JavaScript) and can be deployed to all supported platforms with no need to write native, device specific code or use custom development environments (IDE).
- **Usage reporting** – View reports on how many users are active on the system, what each one is doing, what devices they are using, where they are located (if enabled) and which applications are the most popular.
- **Customizable look and feel** – The AppWorks Mobile Runtimes can be branded to fit an organization’s needs; the name, icon, splash page, login screen and landing page are all configurable. The applications themselves are completely controlled by the application developer and can match whatever experience guidelines the customer has in place.
- **Seamless application updating** – Applications can be updated on the server and seamlessly pushed out to all connected devices without any interaction required by the end user. If a new application is added and enabled by the administrator, that application will appear on the landing page for all users with permission to use that application automatically.
The OpenText Common RESTful API

The Common RESTful API for OpenText Products means that API calls for OpenText Products are made available to developers in a common syntax and with supported common authentication, allowing them to build applications which utilize, update and contribute information to in place EIM Systems with a minimum of effort.

The tenants of the OpenText RESTful API standard are to:

- Focus on adoption
- Keep it simple
- Don’t over think it
- Be consistent
- Always be pragmatic

This section outlines some of the basics of the OpenText RESTful API standards, which AppWorks Developers can utilize to access available OpenText Products through the AppWorks Gateway from a single URL with a single API leveraging a single source for common services.

This section contains a basic overview of the tenants, syntax and some examples of the OpenText Common RESTful API provided by the AppWorks Gateway. The OpenText Common API Standard is subject to change at the discretion of OpenText.

API Case Considerations

URLs

URLs will be all lowercase with no underscores, dashes, or camelCase. The query portion of the URL will be snake_case.

Parameters

All parameters will be snake_case.

Responses

All response fields will be snake_case.
**URL Structure**

A URL identifies a resource. A noun is the only item that is valid within a RESTful URL.

<table>
<thead>
<tr>
<th>URL</th>
<th>METHOD</th>
<th>NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>/dogs</td>
<td></td>
<td>Valid noun</td>
</tr>
<tr>
<td>/GetAllDogs</td>
<td></td>
<td>RPC style. Verbs violate the standard</td>
</tr>
</tbody>
</table>

**NOTE:** Any URLs containing a verb prior to the '?' are non-compliant.

**URL types**

There are only 2 types of URLs for a given resource. One URL interacts with the **collection**, and another URL interacts with the **element**.

<table>
<thead>
<tr>
<th>URL</th>
<th>METHOD</th>
<th>NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>/dogs</td>
<td>GET</td>
<td>This returns a <strong>collection</strong> of dogs.</td>
</tr>
<tr>
<td>/dogs/1234</td>
<td>GET</td>
<td>This returns the full content of the dogs <strong>element</strong> identified by the given key of 1234</td>
</tr>
</tbody>
</table>

**NOTE:** All collections need to be pluralized.

**Associations**

Associations are a way to drill down within an element to an underlying collection.

<table>
<thead>
<tr>
<th>URL</th>
<th>METHOD</th>
<th>NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>/owners/123/dogs</td>
<td>GET</td>
<td>For a specific owner, get all of his dogs</td>
</tr>
<tr>
<td>/owners/123/dogs/1234</td>
<td>GET</td>
<td>This returns the full content of the dogs <strong>element</strong> identified by the given key of 1234 for owner identified by key 123.</td>
</tr>
</tbody>
</table>
Filtering
Filtering allows you to further drill down within a collection and return only the items matching the filter. To avoid name collisions the field you are filtering on needs to have a where_ prefix added to the field.

Limiting which fields are returned by the API
The ability to select and chose the returned fields goes a long way in letting the API consumer minimize network traffic and speed up their own usage of the API. Use a fields parameter that takes a comma separated list of fields to include in the API result.

Sorting Results
To sort the results, you can use the sort parameter. A prefix (asc_. desc_) will specify the direction. If nothing is specified, asc is implied.
Filtering

Often a result set will contain an entry that the end user may want to see the contents expanded. This would avoid an extra API call per fetched row. An example would be an element in a dogs collection where the owner’s info section can optionally be expanded. To expand the contents in a result, use the `expand` parameter.

<table>
<thead>
<tr>
<th>URL</th>
<th>METHOD</th>
<th>NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>/dogs/1234?fields=owner,name</td>
<td>GET</td>
<td><code>{   &quot;dogs&quot;: [     {       &quot;owner&quot;: {         &quot;name&quot;: &quot;Robert Laird&quot;,         &quot;url&quot;: &quot;/owners/2134&quot;       }     }     ]   }</code></td>
</tr>
<tr>
<td>/dogs/1234?fields=owner,name&amp;expand=owner</td>
<td>GET</td>
<td><code>{   &quot;dogs&quot;: [     {       &quot;name&quot;: &quot;buster&quot;,       &quot;owner&quot;: {         &quot;name&quot;: &quot;Robert Laird&quot;,         &quot;address&quot;: &quot;123 main st&quot;,         &quot;age&quot;: &quot;21&quot;,         &quot;location&quot;: &quot;waterloo Ontario&quot;       }     }     ]   }</code></td>
</tr>
</tbody>
</table>

Pagination

Every API call needs to support pagination when calling against a collection. The term `limit` will determine the maximum number of items to return. The terms `before` and `after` will determine the direction.

<table>
<thead>
<tr>
<th>URL</th>
<th>METHOD</th>
<th>NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>/dogs?after=0&amp;limit=20</td>
<td>GET</td>
<td>Get all the dogs in the collection, up to a max of 20, starting at the next item after 0.</td>
</tr>
<tr>
<td>/dogs?after=2&amp;limit=20</td>
<td>GET</td>
<td>Get all the dogs in the collection starting at the next item after 2.</td>
</tr>
<tr>
<td>/dogs?before=4&amp;limit=20</td>
<td>GET</td>
<td>Get all the dogs in the collection starting at the next item before 4 and going backwards.</td>
</tr>
</tbody>
</table>
Versioning
Every API must contain a version string as part of the URL.
The version must be as far to the left within the URL as possible.
The version always contains the letter v followed by the current version as a whole number.
EXAMPLE: /v1/dogs/1234

Verbs
In this section, we will cover which verbs (actions) are required to produce a complaint API.

Verb Basics
The 4 HTTP verbs that are mandatory to support are: POST, GET, PUT, and DELETE. A 5th optional verb is PATCH (used to partially update a resource).

These 4 verbs align with the standard CRUD functions shown below:

<table>
<thead>
<tr>
<th>CRUD</th>
<th>VERB</th>
</tr>
</thead>
<tbody>
<tr>
<td>CREATE</td>
<td>POST</td>
</tr>
<tr>
<td>READ</td>
<td>GET</td>
</tr>
<tr>
<td>UPDATE</td>
<td>PUT</td>
</tr>
<tr>
<td>DELETE</td>
<td>DELETE</td>
</tr>
<tr>
<td>PARTIAL UPDATE</td>
<td>PATCH</td>
</tr>
</tbody>
</table>

POST
POST is the only acceptable way to create a new object within a collection. Exception: In systems where you can specify an ID as part of the object creation, PUT is valid.

<table>
<thead>
<tr>
<th>URL</th>
<th>METHOD</th>
<th>NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>/dogs</td>
<td>POST</td>
<td>On success, a new dog is added to the dogs collection.</td>
</tr>
<tr>
<td>/dogs/1234</td>
<td>POST</td>
<td>Error. This method is not allowed.</td>
</tr>
</tbody>
</table>
On success, the response must contain the structure of the new element created.

**Optional tip:** Create a location header as well.

Must support fields passed on the URL and in the body of the post.

**GET**

GET is only ever used for reads. A GET can never alter the element or collection.

<table>
<thead>
<tr>
<th>URL</th>
<th>METHOD</th>
<th>NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>/dogs</td>
<td>GET</td>
<td>This returns a collection of dogs.</td>
</tr>
<tr>
<td>/dogs/1234</td>
<td>GET</td>
<td>This returns the full content of the dogs' element identified by the given key of 1234</td>
</tr>
</tbody>
</table>

**Fetching a Binary Object**

We need a way to differentiate between a metadata fetch and a content (binary) fetch. Below is how the Tempo Box team addressed the issue.

<table>
<thead>
<tr>
<th>URL</th>
<th>METHOD</th>
<th>NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>/v4/nodes/123454</td>
<td>GET</td>
<td>This returns the metadata about node 12354.</td>
</tr>
<tr>
<td>/v4/nodes/123454/contents (optional extension)</td>
<td>GET</td>
<td>This returns the actual document.</td>
</tr>
</tbody>
</table>

**PUT**

PUT ideally contains the full object representation. On systems where the optional verb PATCH is supported, the expectation is the full object representation is provided. On success, the response must contain the entire structure of the newly modified element.

<table>
<thead>
<tr>
<th>URL</th>
<th>METHOD</th>
<th>NOTES</th>
</tr>
</thead>
</table>
| /dogs/1234?name=fido | PUT     | Renames a dog.
**DELETE**

DELETE is the only acceptable way to remove an item from your system.

<table>
<thead>
<tr>
<th>URL</th>
<th>METHOD</th>
<th>NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>/dogs/1234</td>
<td>DELETE</td>
<td>Deletes dog 1234 from the dogs collection.</td>
</tr>
<tr>
<td>/dogs</td>
<td>DELETE</td>
<td>Deletes the entire dog collection.</td>
</tr>
</tbody>
</table>

**PATCH**

PATCH is the only acceptable way to partially update an element.

<table>
<thead>
<tr>
<th>URL</th>
<th>METHOD</th>
<th>NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>/dogs/1234?name=fido</td>
<td>PATCH</td>
<td>Renames DOG 1234 to fido.</td>
</tr>
</tbody>
</table>

**Handling Limited Verb Clients**

Some clients can only successfully call GET. Some can only successfully call GET and POST. We will support these clients through the use of the method parameter. The responses from these calls should be identical to the response obtained by making the call in a traditional fashion.

<table>
<thead>
<tr>
<th>URL</th>
<th>METHOD</th>
<th>NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>/dogs/1234</td>
<td>DELETE</td>
<td>Deletes dog 1234 from the dogs collection.</td>
</tr>
<tr>
<td>/dogs</td>
<td>DELETE</td>
<td>Deletes the entire dog collection.</td>
</tr>
</tbody>
</table>
Response Formats
The only mandatory response format for OpenText is JavaScript Object Notation (JSON). If an API developer has a business case that demands XML or any other format, the service must support BOTH of the following:

1. Adding the format as a suffix to the collection/element.
2. Specifying the response format using the accept header.

If nothing is specified, the default response will be JSON.

```
/v1/dogs.xml?
/v1/dogs/1234.xml?
```

```
/v1/dogs/1234.xml?
/v1/dogs/1234.xml?
```

```
accept: text/xml
```
AppWorks and the Future

Apps in the context of the enterprise have evolved a lot over the past couple of years. When mobility first became important to organizations, we saw a prolific shift in how enterprise mobile solutions were provided to the market. Security came to the forefront, BYOD changed the way IT considered the concept of ‘Enterprise Standards’ and enterprises and software vendors immediately shifted to the development of specific apps for specific devices, coupled to specific software environments.

This meant that each enterprise system, whether it was EIM, ERP, CRM or other required its own discrete mobile interface and at that; one app for each device type, and this was all supplemental to the existing web and desktop experiences that were fundamental requirements of user acceptance. This was a very expensive and time-consuming way to develop mobile solutions and a very unwieldy and costly to manage architecture for IT departments. This became even more problematic with the increasing trend towards a “bring your own device” workplace. IT could not keep track of all the devices in use and could not create a standardized experience across all platforms and operating systems. To make matters worse, users were frustrated with the mobile experience. Things were not as seamless and easy as they should have been.

Fast forward to the present, and we are making great strides in enterprise application technology. Organizations are implementing platform standardization, while accommodating the BYOD era. OpenText has a unique technology in our AppWorks platform which allows us to author an app once using simple and broadly adopted technologies like HTML, CSS, Java and JavaScript, and then push apps out to all web browsers, desktop users and mobile devices simultaneously and instantaneously; including smartphones, tablets and even kiosks and display panels without additional work for developers or costs incurred by the customer. You create the app once, and then it is available everywhere and it’s integrated with your Enterprise directory server, security policies and corporate systems while providing the ability to leverage the capabilities of any device the app is run on such as: a camera, an accelerometer, a desktop file system, etc. Additionally, we are now seeing the features, which have made consumer apps so popular and successful, work their way into the enterprise.

The demand is to have simple, pleasing enterprise apps for business functions such as onboarding, managing vendor relations, employee files, business processes, etc. These apps eliminate the need to send users to multiple, complex back office interfaces and deliver an integrated, secure user experience that’s easy for IT to manage and extraordinarily simple and compelling for end users.

Lastly, enterprise application management and mobile device management are becoming a larger part of the enterprise mandate. App management allows control of the audience to which your enterprise apps are deployed, enabling you to update apps seamlessly and to install and deploy new apps simply in seconds without any user intervention. It doesn’t end there, the ability to report on and understand app utilization, user uptake, device types, geographic location (if enabled) and other metrics, gives you a view into your workforce never before possible.
Mobile device management provides the ability to manage and secure all enterprise devices effectively; like being able to wipe a device remotely in the event of loss or theft, and the ability to generate reports and gather metrics on which systems are being used the most.

The future of enterprise apps is bright. It’s a world where compelling, effective and ubiquitous apps, which are powered by your existing IT investments, are made available to your audience in a secure, managed and integrated way. The future is here, and it allows you to transform your business by providing the most efficient, adoptable, integrated and secure experience for users, while making it easy and fast for developers to build the apps that drive your business, unleashing the power of your enterprise information.

Contact OpenText

General Inquiries
International: +800-499-6544
North America: +1-519-888-7111

Sales
International: +800-4996-5440
North America: +1-800-499-6544
E-mail: sales@opentext.com

Education
Toll Free: +1-888-673-6839 ext. 2343
Local: +1-519-888-7111 ext. 3
E-mail: training@opentext.com
Offices: office locations

Events
Fax: +1-519-888-0677
Contact: events@opentext.com

Industry Analyst Relations
Phone: +1-519-888-7111 ext. 2410
Contact: Margaret Dobbin

Investors
Phone: +1-519-888-7111 ext. 2408
Email: investors@opentext.com
Contacts: Greg Secord

Media Relations
Kasey Hoffman
Phone: +1 415 500 9551
E-mail: kholman@opentext.com
Stephanie Fazio
Phone: +1-416-464-8414
E-mail: sfazio@opentext.com

Partners
North America: +1-519-888-7111
International: +44 (0) 1189 848 000
E-mail: Global Partner Program

www.opentext.com
NORTH AMERICA  +800 499 6544  ■  UNITED STATES  +1 847 267 9330  ■  GERMANY  +49 89 4629-0
UNITED KINGDOM  +44 0 1189 848 000  ■  AUSTRALIA  +61 2 9026 3400