Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle’s products remains at the sole discretion of Oracle.
### Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introducing Oracle SOA Suite 12c</td>
<td>1</td>
</tr>
<tr>
<td>Cloud Integration</td>
<td>2</td>
</tr>
<tr>
<td>Oracle Cloud Adapters</td>
<td>2</td>
</tr>
<tr>
<td>Cloud Adapter SDK</td>
<td>5</td>
</tr>
<tr>
<td>Mobile Enablement</td>
<td>6</td>
</tr>
<tr>
<td>Internet of Things (IoT)</td>
<td>9</td>
</tr>
<tr>
<td>Productivity Improvements</td>
<td>9</td>
</tr>
<tr>
<td>Rapid Install</td>
<td>9</td>
</tr>
<tr>
<td>Unified Design Interface</td>
<td>10</td>
</tr>
<tr>
<td>Oracle Service Bus</td>
<td>10</td>
</tr>
<tr>
<td>Oracle Event Processing</td>
<td>13</td>
</tr>
<tr>
<td>Templates &amp; BPEL subprocesses</td>
<td>14</td>
</tr>
<tr>
<td>Debugger &amp; Tester Enhancements</td>
<td>16</td>
</tr>
<tr>
<td>New Adapters for On-Premise Integration</td>
<td>18</td>
</tr>
<tr>
<td>Oracle Adapter for SAP R/3</td>
<td>18</td>
</tr>
<tr>
<td>Oracle Adapter for JD Edwards World</td>
<td>20</td>
</tr>
<tr>
<td>Coherence Adapter</td>
<td>21</td>
</tr>
<tr>
<td>Oracle Adapter for MSMQ</td>
<td>21</td>
</tr>
<tr>
<td>Oracle Adapter for LDAP</td>
<td>22</td>
</tr>
<tr>
<td>The Enhanced UMS Adapter</td>
<td>23</td>
</tr>
<tr>
<td>Other new key features in Adapters</td>
<td>24</td>
</tr>
<tr>
<td>Data Translations &amp; Transformations</td>
<td>25</td>
</tr>
<tr>
<td>Feature</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Native XSD (nXSD)</td>
<td>25</td>
</tr>
<tr>
<td>XQuery</td>
<td>25</td>
</tr>
<tr>
<td>EXtensible Stylesheet Language (XSLT)</td>
<td>26</td>
</tr>
<tr>
<td>Management Improvements</td>
<td>31</td>
</tr>
<tr>
<td>Dashboards, Instance Tracking and Error Hospital</td>
<td>31</td>
</tr>
<tr>
<td>Fault Notification Alerts</td>
<td>35</td>
</tr>
<tr>
<td>Performance Tuning</td>
<td>36</td>
</tr>
<tr>
<td>Enterprise Scheduler Service</td>
<td>36</td>
</tr>
<tr>
<td>Continuous Integration</td>
<td>37</td>
</tr>
<tr>
<td>Additional Enhancements</td>
<td>39</td>
</tr>
<tr>
<td>Encrypt/Decrypt Personally-Identifiable Info (PII)</td>
<td>39</td>
</tr>
<tr>
<td>Fault Policy Editor in BPEL</td>
<td>39</td>
</tr>
<tr>
<td>SOA Design-Time Meta Data Services Repository</td>
<td>40</td>
</tr>
<tr>
<td>Startup acceleration through modularity profiles and lazy loading</td>
<td>40</td>
</tr>
<tr>
<td>Re-sequencer in Service Bus</td>
<td>41</td>
</tr>
<tr>
<td>Managed File Transfer</td>
<td>42</td>
</tr>
<tr>
<td>Oracle SOA Suite for healthcare integration</td>
<td>43</td>
</tr>
<tr>
<td>Summary</td>
<td>46</td>
</tr>
</tbody>
</table>
This whitepaper describes how Oracle SOA Suite 12c builds on earlier versions with specific features that address these requirements. In addition, it will discuss product improvements aimed at developer productivity and also introduces a new product, Oracle Managed File Transfer. It is intended for someone who is already familiar with Oracle SOA Suite 11g. If you are new to Oracle SOA Suite, please see the “Oracle SOA Suite 12c – A Detailed Look” white paper for an introduction to the product.

**Introducing Oracle SOA Suite 12c**

Increased integration complexity may seem inevitable as organizations are suddenly faced with the requirement to support three new integration challenges:

- **Cloud Integration** - Rapidly integrate a growing list of cloud applications with existing applications
- **Mobile Integration** - Mobile enable enterprise applications and services
- **IoT Integration** - Connect Internet of Things (IoT) devices to existing systems

To address these integration challenges, companies might assume they need specialized platforms to handle the requirements unique to each. For example, a cloud integration platform to handle heightened security and latency concerns that are of increased importance when integrating cloud applications. Likewise, a separate project within the same company might be looking into a mobile integration platform with support for lighter weight integration standards (REST/JSON, etc.) than traditional enterprise service integration. And finally, another division within the company might be exploring an IoT platform to support the IoT-specific high-volume, high-velocity integration requirements such as advanced proactive filtering and processing not typically included within integration platforms. Over time, these organizations are likely to create an "accidental architecture" of overlapping, disparate, and often proprietary integration platforms resulting in dramatically increased complexity.

This scenario is why Oracle SOA Suite 12c was developed: to simplify integration by unifying the disparate requirements for the latest integration challenges of mobile, cloud, and IoT into one unified and standards-based integration platform.

*This scenario is why Oracle SOA Suite 12c was developed…to simplify integration by unifying the disparate requirements of mobile, cloud, and IoT into one unified and standards-based platform.*
Oracle SOA Suite 12c speeds up your integration time-to-market with features like templates, cloud integration adapters, and mobile-ready integration. You will respond to new business requests more quickly thanks to the new seamless integration of Oracle Service Bus into Oracle SOA Suite, a new managed file transfer offering and significantly improved management capabilities. Your total costs for integration should be reduced as a result of many new productivity enhancements including a quick install capability, advanced new business activity monitoring features and more.

Cloud Integration

Integration entails exchanging information between systems to achieve a specific business objective such as synchronization of customer account data for fulfilling orders. The location of an application and its data, whether in your data center or under the auspices of a cloud provider, does not change this basic business need. While many enterprises have committed some level of investment to the cloud, most of these organizations have to deal with on-premise systems in tandem, or fuse data from another cloud app.

Most enterprises have spent years avoiding the data “silos” that inhibit productivity. After decades of locking down critical issues such as interface definitions, governance, reliability, transaction management, exception handling, and transaction monitoring, is it time to reinvent the wheel yet again for the cloud era? Do cloud applications represent a new silo?

Fortunately, with Oracle Fusion Middleware, one set of integration tools can handle all of these integration scenarios, with direct and interchangeable connections to cloud and on-premise systems. Oracle SOA Suite 12c provides a cohesive set of integration capabilities to simplify diverse IT environments. The recently released Oracle Cloud Adapters simplify the integration of cloud applications with your existing infrastructure to allow, for example, your field sales teams to have real-time access to all of your on-premise applications.

Oracle Cloud Adapters

Oracle SOA Suite aims to simplify, accelerate and optimize integrations with various cloud applications by providing a standard based platform for integration that not only enables connectivity, but also lays a strong foundation to address aspects of audits, compliance, security and governance. In early 2014, Oracle began to offer native connectivity and enhanced developer productivity while integrating with SaaS applications through its newly introduced Cloud Adapters. Oracle Cloud Adapter for Salesforce.com is the first of its kind in this portfolio, and more are coming soon, including Cloud Adapters for RightNow (Oracle Service Cloud), Oracle Sales Cloud, Eloqua (Oracle Marketing Cloud) and Big Machines (Oracle CPQ Cloud). These cloud adapters build on the SOA integration platform to enable standards based connectivity to cloud based applications from on-premise, legacy and other cloud applications, while significantly simplifying the overall life-cycle and user experience. They shield the integration modeler from hand-coding and configuring dedicated logic for handling connectivity, security, and session management individually for each cloud application being integrated with. They also avoid the need for in-depth expertise on the complex functional and technical knowledge of the applications. Thus, with these adapters addressing all the requisites for managing integration with the applications, developers can focus on building the business logic for the integration and business processes.

The cloud adapters enable seamless and simplified connectivity with SaaS applications through their intuitive design-time wizards and rich processing options. In contrast to exposing complex WSDL interfaces for the original service, the cloud adapter configuration wizard engages users with an extremely simplified view of the business object catalog in the application from where they can browse and select one or more objects or services of interest.
for executing the supported operations. The adapter supports integrating with and also visually distinguishes standard business objects from custom objects within the applications.

Several cloud applications also support native query languages used to retrieve one or more business objects and their attributes using a given search criteria. Oracle cloud adapters provide a design-time query editor to build, validate and test these queries right at design-time. This helps shorten test cycles when it comes to integrating with these applications. In addition to provide syntax highlighting for these queries in the editor, the adapters also plan to support real-time code insight capabilities in order to assist the user auto-complete complex queries.
Figure 2: The Query Editor within the Adapter Design-time also provides a design-time test utility to validate the queries modeled by the User

Several SaaS applications impose a cap on the number of user requests that can be made to it on a daily basis. This requires organizations to smartly manage sessions established with the cloud application at runtime. To counter this challenge, Oracle cloud adapters provide intelligent session management capabilities wherein authenticated sessions can be re-used across multiple invocations of the SaaS application across multiple SOA applications.

The most time-consuming component of Integration is data mapping. Cloud applications such as Salesforce.com and RightNow expose complex polymorphic WSDLs that further elevate the costs and complexity of developing data mappings. Oracle Cloud Adapters alleviate this complexity by exposing a strongly typed version of these WSDLs, so that the user can graphically visualize and build data maps with greater ease. The adapter shields this complexity from the user and undertakes the responsibility of translating the generated document into a format, which the SaaS application can understand.

Different SaaS applications have their own unique security requirements for integrations. The cloud adapters expose consistent and simplified security capabilities to integrate with the target SaaS application. For instance, the Cloud Adapter for Salesforce.com enables SSL communication with the cloud applications to prevent eavesdropping while the message is in transit. Furthermore, it also manages the credentials used to communicate with the application within a secure credential store using the WebLogic credential store framework.

In a nutshell, most of the nuances of integrating with SaaS applications such as session management, handling a complex WSDL and security are addressed within the adapter itself. Users are not exposed to these complexities and instead, can focus on fulfilling the business requirement at hand. With all these tasks delegated to the adapter, the possibility of manual errors is reduced, development cycles are cut short and maintenance costs are lowered.
Cloud Adapter SDK

Along with the cloud adapter for Salesforce.com, Oracle SOA Suite 12c also includes a new Cloud Adapter SDK. This SDK allows customers and partners to rapidly on-board new SaaS applications into their enterprise business processes. The Salesforce adapter and all other upcoming adapters from Oracle are developed using the cloud adapter SDK. This SDK is available to partners and customers to develop connectivity and accelerate integration with SaaS applications of interest in a consistent and simplified manner.

The SDK provides a design-time and a runtime component. Design-time components provide various capabilities such as:

- **Connection API** – Define the connectivity to the SaaS application as required by the application.
- **Metadata Browser** – Enable the design-time to browse the metadata within the application.
- **Query Support** - Provide support for creating queries (e.g. SOQL for Salesforce.com, ROQL for RightNow etc)
- **Artifact Generation** – Generate the required JCA and WSDL files for the adapter interaction.

The Runtime APIs provide various runtime capabilities including the following.

- **Session API** – Define the session management capabilities for the adapter
- **Security API** – Security configuration including integration with WebLogic’s credential store framework
- **Transformation API** – Ability to transform the message before sending to the target SaaS application in case the design-time data definition is different from the expected data definition. This is often the case for complex polymorphic data definitions for which a simplified structure (XSD, WSDL) is defined at design-time for easier modeling.
- The design-time APIs are plugged into the adapter configuration wizard within JDeveloper. The runtime APIs are used by the SOA Suite and Service Bus runtime to invoke the target SaaS application as shown in Figure 4.
Mobile Enablement

For many years web browsers and desktop computers were the primary platforms for accessing information from enterprise applications. As mobile devices proliferate, applications are no longer tied to the desktop. Users want to use their smartphones and tablets to access corporate data and business apps, anytime, anywhere. In support of this paradigm shift, Representational State Transfer (REST) and JavaScript Object Notation (JSON) have emerged as the dominant standards for exposing services and APIs for mobile enablement.

When addressing this challenge, mobile should be treated as an extension of the applications that are already in place and developers should simply extend their SOA integration methods and infrastructure rather than adopting new development and deployment infrastructures to address the mobile challenge.

Oracle SOA Suite 12c introduces a REST binding within JDeveloper to simplify mobile enablement by exposing traditional SOAP services, Enterprise Java Beans (EJBs), JCA adapters connecting backend applications or just about any other underlying implementation through JSON/REST. The REST binding is available for SOA composites and Service Bus services and allows the configuration of REST interactions as exposed interfaces and allows the invocation of externally available REST services.

Multiple ways to define a REST binding through a simple wizard are included: the definition can be on an existing WSDL, or the binding can even create the WSDL for you. When using the REST binding, a WADL (Web Application Description Language) file is created automatically to define the REST interaction and to allow interaction with the REST binding throughout the integration layer. Oracle SOA Suite 12c practices a “REST on the edges” approach where a REST/XML, REST/JSON, or URL-encoded payload is translated to and from SOAP/XML in the integration wire. This is called a “translation on the wire”.

Figure 4: Illustration showing the integration between JDeveloper and the SOA Runtime with the Cloud SDK (Plugin)
This approach has been selected because SOAP/XML has been well accepted for many years. Any experienced integration developer is quite familiar with SOAP/XML interactions and will likely want to reuse as much as possible rather than a complete “rip-and-replace”. By translating, you can interact with REST style edge interfaces while leveraging your existing integration layer. This allows an existing SOAP/XML interface to be exposed easily as a REST style interface.

By exposing back-end applications as RESTful APIs through your existing SOA composite or Service Bus, which then connects to backend services and applications, developers are able to leverage existing application and integration infrastructure as part of a highly reusable and scalable development and integration platform for mobile applications, which communicate through Oracle Service Bus as a standardized mediation and virtualization layer.

Figure 5: Oracle Service Bus exposing a backend SOAP service as REST service

Both SOA composites and Service Bus projects include the same REST binding wizard. To avoid duplication, the remainder of this section will discuss REST support within Service Bus.

When exposing an existing SOAP service, the REST binding automatically takes the same shape as defined in the original WSDL. If a different shape needs to be supported, a REST interface can be created based on a different schema and messages can be transformed in the pipeline. Remember, the translation (REST to SOAP to REST) takes place on the wire between the pipeline and the Proxy and/or Business services.

Figure 6 shows how a SOAP based payment service is exposed as a REST service through the new wizard. You can see that the parameters for the REST service are auto-created and can be changed manually. The HTTP Verb dropdown allows the creation of bindings for all of the standard verbs. The wizard allows the definition of the payload type as XML or JSON or both, sample data creation as XML or JSON and HTTP status definition for each operation.
The Service Bus test console has been enriched for easy testing of REST services by providing an input field for the resource to be tested and drop-down menus to select the method the accept header, and whether to display the response as XML or JSON.
Internet of Things (IoT)

As devices become connected and the Internet of Things becomes ubiquitous, the multitude of devices will enable companies to improve customer service, offer newer products or streamline existing processes. Middleware plays a key role in acting as a bridge between such edge devices or things and enterprise applications.

The role of middleware is to provide the infrastructure and IoT services which in turn help drive innovation, enable new revenue streams, and improve operational efficiencies.

Event processing is an integral part of an Internet of Things platform. As massive amounts of data streams from sensing devices, it’s important to derive an understanding of what data is important and what’s not.

**Oracle Event Processing (OEP)** delivers real-time analysis of high-velocity data. It is a complete solution for building IoT applications to filter, correlate and process events in real time so that downstream applications are driven by true, real-time intelligence. OEP filters out noise (such as data ticks without any change in values) and helps identify critical conditions before this data is actually relayed to the back-end. Built on industry standards including SQL, Java, and OSGI, it provides an open architecture for processing complex events. Oracle SOA Suite 12c more tightly integrates the OEP platform with Oracle Service Bus and the Oracle SOA Suite Event Delivery Network (EDN). More information on this important component is available in separate documents.

**Productivity Improvements**

In addition to adding new features supporting emerging industry trends like cloud and mobile, Oracle SOA Suite 12c also provides many new features and enhancements around developer tooling, administration and operations and, of course, performance.

**Rapid Install**

Crucial first steps in every SOA development cycle are the **installation** and **configuration** of a SOA **development environment**, including the Integrated Development Environment (IDE) and the SOA run-time and infrastructure database. For pre-12c versions of Oracle SOA Suite, these steps were challenging, starting with the download of several components and a complicated install and configuration process.

With Oracle SOA Suite 12c, this process has been dramatically simplified with the introduction of the new **Quick Start installer**. The download consists of a single file that can be obtained from the Oracle Technology Network (OTN). The user only needs to answer a few questions, such as defining the location for Oracle Home, and the installation usually completes in less than 30 minutes.

The Quick Start install provides the foundation for a single user development environment, which includes all Oracle SOA Suite core components - Oracle BPEL Process Manager, Oracle Human Workflow, Oracle Business Rules, Oracle Mediator, Oracle Service Bus, Oracle Technology Adapters, Oracle Enterprise Scheduler, SOA Spring Component and Enterprise Manager Fusion Middleware Control. In addition, Oracle JDeveloper is installed with all SOA IDE extensions and an integrated WebLogic Server.
After the installation completes, you only need to create and then start the pre-configured development domain, in this case an integrated WebLogic Server domain in Oracle JDeveloper, which uses Java DB, also known as Apache Derby, for the SOA dehydration store.

At this point, everything is ready to go and you can either download the provided SOA tutorial or start with your first Oracle SOA Suite 12c application.

Unified Design Interface

A key differentiator of Oracle SOA Suite 11g relative to other integration platforms has always been the unified design interface across most of the components for Oracle SOA Suite. Oracle SOA Suite 12c takes this differentiator a big step forward by integrating the remaining major components for Oracle SOA Suite, Oracle Service Bus and Oracle Event Processing, into a single unified experience. This feature simplifies integration by eliminating the need to train developers, administrators, architects and others on separate components for every aspect of integration, which lowers cost and provides a faster time to integrate.

Oracle Service Bus

The visual representation of an Oracle Service Bus project, the Overview editor (see Figure 9), has been modeled from the SOA Composite Editor and now shares the same JDeveloper IDE environment with the rest of Oracle SOA Suite. This view allows the creation of Service Bus projects using a top-down, drag and drop approach. You can create services by dragging icons from the Component Palette on the right, to the lanes of the canvas.
Figure 9: The Service Bus Overview in JDeveloper with a proxy service, pipeline and business service

On the Component Palette seen in Figure 10, notice that the Resources category contains Pipeline and Split-Join icons, the components for a Service Bus application. In this release, the pipeline has been split from the proxy service to allow it to be a re-usable component. The proxy is defined through the transport while the pipeline includes the message flow. Due to this split, a pipeline can now be exposed through several transports, allowing more than one input channel into the same message flow.

The palette also includes adapters and Service Bus transports. While all SOA adapters are now certified with Service Bus, the transports will still be available. In order to minimize confusion, the group Technology includes recommended technology adapters and transports. Others are grouped in Advanced for completeness and backward compatibility.

You can also see Application and Cloud Adapters in the component palette (Figure 10).
Figure 10: The Oracle Service Bus component palette in JDeveloper includes Pipeline, Split-Join, Technology Adapters and Transports, Application Adapters and Cloud Adapters.

Traditionally, Service Bus always provided a browser-based console for light-weight Service Bus configuration. This console is still available in Service Bus 12c and has received a fresher look as you can see in Figure 11.
Oracle Event Processing

With Oracle SOA Suite 12c, OEP has been integrated into JDeveloper, providing a visually impressive component palette to create your applications by simply dragging and dropping the required artifacts onto the canvas. With each event node dropped, wizard driven prompting is shown to ensure that the correct and required parameters are quickly defined. Another major new feature is the Event Delivery Network (EDN) event adapter nodes, which provide that tight integration at runtime to Oracle SOA Suite composites and all other SOA components supporting the EDN. Using the resource components window view, the Oracle Event Processing runtime server and application status can be easily reviewed and manipulated effectively.
Templates & BPEL subprocesses

“Share and reuse” has always been one of the pillars of a service-oriented architecture. Oracle SOA Suite 12c introduces several new features to boost the ability to share and reuse services and components, while at the same time enabling best practices and improving manageability. The new SOA starter templates provide an easy way to package SOA composites, components or fragments of a BPEL process and distribute them to other departments, partners or customers. This enables the sharing of frequently used patterns and speeds up development for template users. Three types of SOA starter templates are supported:

**Project templates** are a new way of creating a SOA project instead of starting with an empty composite. The template can include pre-defined components, as well as services and references. For example, frequently used adapters and transformations can be pre-configured in a template to eliminate the need to re-create the same components every time a new project is created.

**Component templates** package one main component, for example a BPEL process, including all references (adapters and bindings) invoked by the process. You can enforce best practices by packaging a BPEL process with a pre-defined fault handler and a number of mandatory variables as a component template and sharing it across departments. By providing this template to all developers, you ensure a consistent way of handling faults and naming variables.

**Custom activity templates** consist of a group of activities in a BPEL process, which can include partner-links and external services. The templates are available within the BPEL palette and can be dragged and dropped into any BPEL process. You can for example package a BPEL scope, consisting of a complex assign or transformation and the invocation of an external web service. By sharing this scope, you eliminate the chances of introducing errors and of course save development time through re-use.

SOA templates work in a “drag-and-drop” way. Once they have been used, they become an integral part of the application and can be customized going forward. When changes are made to the original template, they’re not reflected in the composites and components created through templates.

Note that pipeline templates in Oracle Service Bus fulfill a slightly different use case. Many customers have been asking for a faster way to provision services that are very similar to each other, apart from small differences, like a different routing end point or service call-outs. The template author defines the message flow and marks those areas allowing customization as so called “placeholders” (see Figure 13). Whenever a new “derived” pipeline is created based on a template, only the areas defined as placeholders can be changed. All other stages and actions within the pipeline will be preserved as long as the pipeline stays connected with the template. Derived pipelines inherit all future changes to the template.
If the connection between a derived pipeline and a template is intentionally broken, the pipeline behaves like any other pipeline and changes will not be inherited anymore.

Both types of templates – SOA starter templates and pipeline templates - accelerate development, reduce the chance of errors when duplicating patterns and still provide enough freedom for customization. At the same time, there is no difference at runtime between services or components generated from scratch or those derived from templates.

**BPEL subprocesses** also promote re-use and sharing, but work slightly differently. Don't be fooled by the name, this is not just a fancy new way of calling one BPEL process from another. A BPEL subprocess is actually not a complete process, but a fragment that modularizes a reusable piece of business logic. This fragment can include partnerlinks and references and can be invoked from other “parent” BPEL processes through a new “call” activity. A subprocess doesn’t have an interface, like a standard BPEL process does. You can best compare it to a subroutine in an object-oriented programming language. At run-time, it executes completely in the context of the parent process, permitting access to process variables and inheriting fault handling and compensation logic.

Oracle SOA Suite 12c provides two types of BPEL subprocesses: an **inline** subprocess is implemented as part of a parent BPEL process and only visible to this one process. It can be invoked anywhere within the parent. A **standalone** subprocess can be invoked by all BPEL processes within the same composite and has a visual representation in the composite editor. You can see in Figure 14 that the wire between the parent and the subprocess is represented as a dotted line to visualize the difference to a normal wire in a composite.
Like a template, a BPEL subprocess enforces best practices and reduces development time by eliminating the need to re-create the same activities over and over again. A BPEL subprocess however also boosts performance: It takes up less memory at run-time as it only gets created in memory once, independent of the number of invocations.

**Debugger & Tester Enhancements**

A successful development process requires simple tools for debugging and testing during development. Oracle SOA Suite 12c includes a visual debugger in Oracle JDeveloper that allows the setting of breakpoints within a SOA composite, a BPEL process and a Service Bus pipeline. For composite breakpoints, you have a choice between request and reply breakpoints or a breakpoint pair. Figure 15 shows a BPEL process with a SOAP interface and two breakpoint pairs. The debugger will stop four times: after the SOAP requests is received, before the BPEL process starts, after the BPEL process completes and before the SOAP response is returned.

**Figure 14: Standalone subprocess in the component palette including database adapter shown as dotted line**

**Figure 15: Breakpoints in a SOA composite highlighted by arrows**
Within a BPEL process, breakpoints can be set for every activity.  

![Figure 16 Setting a breakpoint in a BPEL process](image)

The debugger provides stepping into, out and over, just like a Java debugger, and provides you with a data window that shows all currently visible variables and their values. Variable values can be changed during the debug process. Debugging is only allowed in development environments, on local or remote servers, but not in production systems.

![Figure 17: Data window during debug](image)

The **SOA Suite test framework**, already available in previous releases, has been enhanced in Oracle SOA Suite 12c. As part of a test definition, input and output messages can be auto-generated or loaded from an existing sample file. Request and response messages as well as faults can be asserted and external services and faults can be emulated to avoid testing of backend systems. The actual test can be run directly within JDeveloper, which also shows detailed reports of every test run. There’s no need anymore to switch between JDeveloper and Enterprise Manager Fusion Middleware Control to define, run and analyze tests.
Figure 18: Test definition in JDeveloper

Figure 19: Test Results in JDeveloper

New Adapters for On-Premise Integration

**Adapter** are the critical last mile components that simplify integration across enterprise applications, legacy and homegrown systems, and now also allow integration with cloud applications. Several new on-premise **Application** and **Technology Adapters** are available for Oracle SOA Suite 12c.

**Oracle Adapter for SAP R/3**

Along with Oracle SOA Suite 12c, Oracle introduces the new SAP Adapter that provides native bidirectional integration with SAP R/3. It supports invocation of BAPI/RFC and IDOCs to send and receive data from the SAP R/3 application. The Adapter is natively integrated into JDeveloper and communicates with the SAP Business Object Repository to provide a graphical browser of the objects in SAP for the user to discover, search and select for the integration. In addition to browsing the objects in the repository, the adapter design-time allows searching for the objects by name or wild card expressions.
The design-time wizard provides great detail on each of the objects, such as the detailed view of the definition and the corresponding XML Schema. Most importantly, it enables users to test the APIs in SAP R/3 at design-time, without needing to deploy the SOA application to the application server.

Figure 20: Wildcard search for BAPI objects in the SAP Adapter design-time

Figure 21: Testing the BAPI/RFC APIs at design-time
Oracle Adapter for JD Edwards World

The JDE World Adapter enables users to natively integrate with JD Edwards World ERP systems through a JDeveloper based design-time that enables users to perform integration with the JDE World ERP Application. In Oracle SOA Suite 12c, the adapter supports insert and query of data from the JDE World z-tables.

Figure 22: Viewing BAPI Execution Test Results at design-time

Figure 23: The System Browser in the JD Edwards World Adapter enables users to search, browse, discover and integrate with z-tables in JDE World.
The adapter uses the JDBC Driver for JDE World for connecting to the Application. Note that it requires availability of jt400.jar library to access the JDE/AS/400 system that serves as the foundation for the ERP application.

Coherence Adapter

The Coherence Adapter allows seamless integration with Oracle Coherence, the industry leading in-memory data grid solution that enables organizations to predictably scale mission-critical applications by providing fast access to frequently used data. In many cases, data in backend applications or databases doesn’t change very often. Retrieving data like this by directly accessing these systems every time is too costly. The coherence adapter provides the ability to push the data to the coherence cache and access it directly from the cache when needed. The adapter supports outbound integration with local and remote coherence cache and allows users to insert, retrieve, remove and query XML and POJO data from coherence. Users can use the coherence query language to filter and query data from the cache.

![Image](image.png)

**Figure 24:** The Cache Query configuration in the Coherence Adapter

Oracle Adapter for MSMQ

The messaging adapters also got a new addition in Oracle SOA Suite 12c - the **MSMQ adapter**. MSMQ is the messaging middleware from Microsoft, running on the Windows operating system. The adapter allows you to send and receive messages from private, public queues and Distributed Lists. It uses WebLogic jCOM to communicate with the MSMQ Server and exchange messages.

If the MSMQ Server and SOA are on the same Windows machine, the adapter provides superior performance by using a native mode of communication.
Oracle Adapter for LDAP

The **LDAP adapter** provides bi-directional integration with several LDAP V3 directory servers and enables users to perform real-time query, CRUD, compare, search and also issue DSML requests to the server (outbound). It also enables business processes to be triggered on changes to data in the directory server (inbound). The LDAP adapter support the following operations:

» Inbound: Change log notification, Entry change notification
» Outbound: Add, Delete, Modify, Modify DN, Compare, Search, Execute a DSML Request

It also has a rich LDAP Browser built in at design-time that enables users to configure the search requests accordingly at design-time.
The Enhanced UMS Adapter

The UMS Adapter was introduced in Oracle SOA Suite 11.1.1.7 with bidirectional support for email communication - i.e. enable email communication from SOA composites, as well as ability to trigger SOA composites on arrival of emails. The adapter has been enhanced to enable bidirectional communication with additional messaging channels such as SMS and instant messaging.
Other new key features in Adapters

Cross-Platform Support

From Oracle SOA Suite 12c forward, all adapters will be supported within SOA Suite, Service Bus and BPM projects.

Scheduled Activation and Deactivation

Thanks to the addition of the new Enterprise Scheduler Service (see Enterprise Scheduler Service), polling adapters can now be restricted to certain times of the day, for example outside of business hours, to avoid resource overload. Users can associate an inbound adapter for a SOA Service with ESS schedules to customize timings for processing of messages.

Debugger Integration

Users can enable debugging at the adapter binding components to troubleshoot issues. The debugger enables users to view native and translated data from the adapter for inbound and outbound interactions.

Monitoring and Diagnostics

The Enterprise Manager Fusion Middleware Control now provides diagnosability reports for all adapter binding components for SOA applications. These reports provide real-time visibility into availability and message exchange statistics with the Application or endpoint. The three new report types are Configuration Reports, Monitoring Reports and Snapshot Reports.

Figure 28: New Fusion Middleware Control reports in Oracle SOA Suite 12.1.3
Data Translations & Transformations

**Data** is at the heart of every integration project. It can be received and transmitted through a number of different protocols and many different shapes, for example XML, JSON, comma separated lists and many more.

Native XSD (nXSD)

In previous versions of Oracle SOA Suite, adapters provided the ability to **translate** messages from non-XML formats to XML and vice versa through native XSD (nXSD). Only XML could be processed within a SOA composite. Oracle SOA Suite 12c extends this ability to use nXSD translation as a first class citizen within a BPEL process, Service Bus and Mediator. Now messages can be translated anywhere within the integration flow, not only on the edges. Data **transformation** from one XML format to another can be performed through either **Extensible Stylesheet Language (XSLT)** or **XQuery**. Oracle JDeveloper now includes graphical mappers for both transformation types.

XQuery

The new **XQuery Mapper** provides a bi-directional view to construct XQuery 1.0 modules and libraries. The left pane of the XQuery Mapper shows the source tree, and the right pane shows the target tree. Tree nodes can be XML elements, attributes, and some other XQuery constructs. In addition to the mapper design view and the source view, expressions can also be edited in the properties window. The properties window can also be used to create more complex XQuery expressions either from pre-defined functions or by defining your own.

![XQuery Mapper in JDeveloper](image)

Figure 29: XQuery Mapper in JDeveloper
Oracle SOA Suite has always provided a graphical XLST Mapper. In the 12c release, this mapper has been enhanced to support large and complex stylesheets in the design view without the need to switch to source view through the introduction of two different views: Map View and XSLT View.

Map View is the traditional view of the XSLT mapper that was available in previous releases and can be used for simpler maps. In Map View the user sees a simple single template rule that contains all the XSLT instructions to be executed. Possible output nodes from the target schema are displayed under the template for easy drag and drop mapping. The user may also add simple XSLT instructions to create loops and conditional statements.

XSLT View is a more advanced mode that supports several new features, including Named Templates, Template Rules, Execution View, Completion Status and Testing.
The XSLT view is intended for the advanced user who would normally edit XSLT in source view. The XSLT View includes the same panes as the Map View, except that the right target pane is divided into two panes. The top pane is called the XSLT pane and the lower pane is called the target pane. If no target schema is defined, then the lower pane is not shown. If no source schema is defined, the source pane is still displayed to enable you to add parameters and variables, whose values can be referenced by the XSLT. The XSLT View enables you to graphically display and edit any XSLT stylesheet, irrespective of the complexity involved.

**XSLT templates** are used for modularity – similar to SOA and pipeline templates – and can be compared to a method in an object-oriented programming language. They allow a single XSLT stylesheet to be broken into multiple logical units, each of which performs a specific transformation. These units can be reused by placing them into XSLT files that can be imported or included into other stylesheets.

**Named templates** can be invoked by calling the template explicitly by name. **Template Rules** are template statements that include “match attributes” to specify a pattern. A template rule executes when the pattern specified by the match attribute matches the node currently being processed by the XSLT processor. This node is referred to as the context node for the template rule and XPath expressions used within that template are usually relative to the context node for the template. The XSLT Map Editor now supports editing of both template rules and named templates in the XSLT View. More specifically, the XSLT View provides a graphical view of all XSLT statements that can **highlight context nodes** for the selected template rule allow the immediate error detection when creating multiple templates.

For example, Figure 33 shows the reuse of a single template rule to process all Item nodes. Note that all Item nodes in the source tree are highlighted. This highlighting indicates that the “match = “Item”” template on the right will process these Item nodes. The absence of highlighting for a defined template can indicate a problem in the definition of that template.
Figure 34 shows context highlighting when using an identity template. An identity template can copy the incoming XML document using only 3 lines of XSLT. We are copying all nodes in the incoming document except for the Item nodes, which will be modified as the rest of the document is copied. The second template in the figure will process the Item nodes, while the first template (the identity template) will process all the other nodes in the document with a simple copy. The highlighting indicates the nodes each template will process.
The **Execution View** shows the order of execution of templates (both named and matched) as they would be executed by the processor. You can see the overall flow of a complex XSLT process and provides a means to troubleshoot problems in template creation before executing the XSLT. In addition, the user can use the execution view to navigate to any section of the stylesheet by clicking on the desired node in the execution view dialog.

Figure 35: Execution View

Figure 36 shows the editing of a named template called “createStreet”. The graphical editor provides the same drag and drop mappings as the Map View, allowing you to rearrange XSLT statements, work on XSLT with or without defining source and/or target schemas, and create reusable template rules. These features make it possible to create complex XSLT much faster and more easily than it would be to edit text in source view.
XSLT View also eliminates the display of possible target schema nodes as are shown in Map View. This allows the user to construct more complex XSLT stylesheets and to work against extremely large schemas more effectively. XSLT View still provides support for inserting target schema elements through a context menu and drag and drop from a target schema tree.

If you are using a target schema for your map, you can check the completion status of the map at any time. The completion status dialog lists all unmapped target elements and attributes, target elements mapped with incomplete XPath expressions, and all missing target elements and attributes.

Figure shows the completion status dialog. Note that the user can navigate to the problem area of the map by clicking the desired line in the dialog.

The XSLT Map Editor provides a tool to test your map by right-clicking the canvas pane, and selecting “Test” from the context menu. You can use the test tool in both Map View and XSLT View. Figure 38 shows the invocation of the test tool.
If you are working with large schemas it can be useful to zoom into a specific portion of the map. The XSLT Map Editor provides a scoped view option to zoom into an area of the mapping. The user can continue to work in this scoped view, shown in Figure 39.

Management Improvements

While the previous sections mainly focused on features improving developer productivity, this section will discuss new features for operations and management.

Dashboards, Instance Tracking and Error Hospital

The main tool for Oracle SOA Suite administrators is Enterprise Manager Fusion Middleware Control. In Oracle SOA Suite 12c, this now also includes the management of Oracle Service Bus projects. The Service Bus 11g console has been divided into two parts: configuration of services is still available in the browser-based console, in addition to the JDeveloper IDE. All operations and management tasks have been moved into Enterprise Manager Fusion Middleware Control to provide a single administration console for the core SOA components.

Enterprise Manager Fusion Middleware Control itself has undergone a number of changes to improve responsivenes and simplify day-to-day administration and troubleshooting.
The main **SOA dashboard** has been redesigned to focus on system health and a consolidated view of faults and issues.

Figure 40 shows five summary sections in one view:

- **Key Configuration** shows the modularity profile, instance tracking value, default query duration and auto purge state with links for more information and ability to change these configuration parameters.
- **SOA Runtime Health** and **System Backlogs** show a quick overview of your composites. The system backlogs are not populated by default. This is a new feature of Oracle SOA Suite 12c to improve performance and responsiveness of Enterprise Manager. The refresh icon populates the backlogs.
- **Business Transaction Faults** is also not displayed until refreshed. The default query duration (as displayed in the key configuration area) can be changed at any time.
- The **Search** section provides quick options for some common searches plus any searches that you have saved. This search will take you to the **Flow Instances** tab.
- The **Fault Alerts** section shows any alerts that have been generated.

![Figure 40: Enterprise Manager Fusion Middleware Control Dashboard](image)

In the past, retrieval of large amounts of data has caused performance problems. In order to improve this, searches have been enhanced to allow for finer grained queries, including composite sensor values, to track specific business cases. Searches can also be saved and bookmarked for quicker access.

To prevent initial overload of a page, the user has to actively choose a search before a list of items is displayed.

The instance tracking experience in Oracle SOA Suite 12c is highly enhanced and not just on usability but also in the underlying architecture to improve performance, visibility, and traceability of your end-to-end transactions, which
now also include Service Bus, B2B and MFT. Figure 41 shows a list of flow instances and the search pane that allows the creation, saving and bookmarking of fine-grained searches.

![Figure 41: Flow Instances](image)

The new Error Hospital in Oracle SOA Suite 12c is used to aggregate flow instances that have generated faults and be able to perform actions on instances with common faults collectively. The aggregations can be based on various criteria based on fault types, names or composite level aggregations.

Some faults can be recovered after the reason for the fault has been fixed. For example, if a database was unavailable, the administrator can recover the faulted instances after the database has been restarted. This recovery can be done one-by-one, or a group of instances can be recovered together in a bulk operation, either immediately or at a later point in time to optimize on resources. Figure 42 shows the dashboard populated with faults. Clicking on the fault area opens the new Oracle SOA Suite 12c Error Hospital with faults, which can be grouped by composite.
Figure 42: SOA Dashboard with Faults

Figure 43: Error Hospital with faults grouped by composite
Fault Notification Alerts

While dashboards provide an overview of the system and application health, administrators need to be alerted automatically when something goes wrong in the system. Fault alerts can be routed to communication channels such as email, SMS and IM, while at the same time being posted to the dashboards. Instead of always sending the alerts immediately, they can be scheduled to trigger an alert based on a predefined error notification rule.

Figure 44: Recurring Schedule for fault notification alerts

Figure 45: Create Notification Rule
Performance Tuning

Oracle SOA Suite 12c makes use of WebLogic Server work managers to optimize scheduled work across SOA components. This simplifies the thread pool configuration and allows SOA to effectively use the existing resources. This also provides self-tuning which gives additional resources to specific components if they are otherwise idle.

For a SOA production system to be performant, the dehydration data store needs to be optimized, including appropriate database partitioning and regular purging of obsolete data. Oracle SOA Suite 12c provides pre-tuned database profiles to automatically enable appropriate performance features, based on the expected data size. It also enables auto-purge to prevent bloating of the database.

Enterprise Scheduler Service

Enterprise Scheduler Service (ESS) is a new component within Oracle SOA Suite 12c, which was previously only available through Fusion Applications. It provides the ability to schedule SOA components or services and manages the complete life cycle of a job definition: development, distribution, scheduling, and monitoring.

![Enterprise Scheduler Service](image)

Figure 46: Enterprise Scheduler Service Job Definition

In addition, polling adapters can be activated or deactivated at specific times to minimize resources during busy hours. For example, an application that checks and re-fills the inventory should run outside of business hours, to not interfere with the daily business.
The Enterprise Scheduler Service is also used for bulk error recovery, as described in the Error Hospital section, and the triggering of fault alerts.

Continuous Integration

Continuous Integration brings agility to the systems development lifecycle (SDLC) by completing packaging, testing and deployment of every change to the code. In order to accelerate development and to reduce the time-to-production of new features, development teams seek ways to automate the build, test and deploy processes. Many development teams have selected Maven for standardizing their projects choosing convention over configuration.

Oracle SOA Suite 12c provides a Maven plugin that allows development teams to use Maven to create, build, package and deploy SOA projects. Through the SOA application and project archetypes, development teams can automate the instantiation of a developer's working environment. Through the Maven plugin goals, a Continuous Integration server such as Hudson can be used to manage SOA projects through the complete lifecycle. Using Maven, teams can increase their agility and accelerate solutions to production with lower cost and higher confidence. Figure 48 shows how the running of Maven phases has been integrated into JDeveloper.
Figure 48: Contextual menu to execute standard Maven build phases

Figure 49 shows a graphical editor to edit Maven POM files. POM stands for "Project Object Model". It is an XML representation of a Maven project held in a file named pom.xml. A project contains configuration files, as well as the developers involved and the roles they play, the defect tracking system, the organization and licenses, the URL of where the project lives, the project's dependencies, and all of the other little pieces that come into play to give code life. It is a one-stop-shop for all things concerning the project.

Figure 49: Graphical editor to edit Maven POM files
Additional Enhancements

Encrypt/Decrypt Personally-Identifiable Info (PII)

Instance tracking within Enterprise Manager Fusion Middleware Control often exposes message payloads to administrators. In most cases that is not a problem, unless the message includes sensitive data, like a social security number. The new PII feature enables the encryption and decryption of specific fields in your message to protect this kind of sensitive data directly in Oracle JDeveloper by applying Oracle Web Services Manager (OWMS) policies.

![Encryption of Personally Identifiable Information](image)

Figure 50: Encrypt Personally Identifiable Information

Fault Policy Editor in BPEL

Oracle SOA Suite provides a fault management framework for handling faults in BPEL processes. If a fault occurs during runtime in an invoke activity in a process, the framework catches the fault and performs a user-specified action defined in a fault policy file associated with the composite or component. Fault policies are applicable to the faults that result from the invoke activity. A fault policy bindings file (fault-bindings.xml) associates the policies defined in a fault policy file (fault-policies) with the SOA composite application or the component. Within the fault policies file, you can define what actions should be taken when a specific fault occurs. In Oracle SOA Suite 12c, fault policies can be configured using the new Fault Policy Editor instead of editing the XML file source.

![Fault Policy Editor](image)

Figure 51: Fault Policy Editor

With the editor you can also configure alerts to notify an administrator or enqueue the fault to a JMS queue or publish it to a JMS topic.
In addition to the existing run-time Meta Data Services Repository, Oracle SOA Suite 12c adds a file-based SOA Design-Time MDS Repository for use during design time. The repository is automatically created when you create a SOA composite application and will typically point to the version control system location.

The SOA Design-Time Meta Data Services Repository provides a wizard to share design time artifacts such as WSDLs and XML Schemas and eliminates the need to duplicate these resources across projects and applications. When artifacts are moved into the repository, all dependencies are included and references to the transferred artifacts are updated with the correct MDS URLs on transfer.

New modularity profiles provide support for starting up the SOA platform in subsets of functionality, reducing the overall memory footprint of the SOA instance, thus reducing the overall startup time of the platform. To achieve this, subsets of SOA functionality are separated into groups. For example, “SOA Foundation”, the default for new domains, reflects the same configuration as exists in Oracle SOA Suite 11g today. “BPEL-Only” is a very lightweight profile that supports composites with BPEL processes, but does not include Mediator or Rules.

A significant difference of up to 30% in footprint has been observed between the largest profile “SOA_CLASSIC” and the smallest “BPEL_ONLY” profile. The profiles can be changed at any time in Enterprise Manager Fusion Middleware Control after the domain has been created.
The second feature in this category reduces the startup time of the SOA platform by **staggered – “lazy” - loading** of composite artifacts and spreading out bootstrap costs. This is especially important for disaster recovery and active-active setups where fast server startup is essential. By loading composites minimally at server startup, overhead from rarely used and retired composites is reduced. The first runtime request for a composite loads the components and resources (Schemas/WSDL/XSLTs etc) fully. When using the lazy loading feature, you need to consider whether a delay in the first invocation of a composite is acceptable.

Re-sequence in Service Bus

Business transactions often span multiple technical transactions and require groups of messages to be processed in **exactly the right order**. Sometimes, these messages already arrive in the correct order and need to be prevented from getting out of order, or they arrive in random order and need to be re-sequenced. The re-sequence in Oracle Service Bus orders messages based on sequential or chronological information, and then sends the messages to the target services in an orderly manner. The sequencing is performed based on the sequencing strategy selected.

The re-sequence works with two central concepts: groups and sequence IDs. The sequence ID is an identifying part of the message, and messages are rearranged based on this identifier. The messages arriving for re-sequencing are split into groups and the messages within a group are sequenced according to the sequence ID. Sequencing within a group is independent of the sequencing of messages in any other group. Groups in themselves are not dependent on each other and can be processed independently.

Groups and sequence IDs are identified through XPath expressions in the message payload and header. You specify XPath expressions that point to the elements in the message payload on which grouping is done and on which sequencing is done. Incoming messages are sequenced based on one of the following strategies:

- **Standard**: The standard re-sequence is useful for applications that use identifiers from a simple numeric identifier sequence in their messages. The standard re-sequence receives a stream of messages that might not arrive in order; it then stores the out-of-sequence messages until a complete sequence based on the sequence IDs is received. The in-sequence messages are then processed asynchronously based on their sequence ID.

- **FIFO**: Sequencing is based on the time of the arrival of the message and supports a standard first in, first out (FIFO) pattern.

- **Best Effort**: The best effort pattern is useful for applications that produce a large number of messages in a short period and cannot provide information to the re-sequence about the identifier to use for sequencing. Typically, the identifier used in such scenarios is of a dateTime type or numeric type. Using the dateTime field as the sequence ID XPath enables you to control the sequencing. The messages are expected to be sent in sequence by the applications, thus the date and time the messages are sent can be used for sequencing. The best effort is made to ensure that the messages are delivered in sequence.
Managed File Transfer

With all of the discussion of Cloud computing, it may be tempting to lose track of the core needs and requirements for a fully integrated file transfer solution. Organizations still make heavy use of files utilizing bulk, batch and real-time updates for application integration. If anything, file integrations look to increase in both size and volume as you incorporate cloud Software as a Service (SaaS) into your IT landscape. You will find that the general approach for on premise and partner file integrations using Managed File Transfer (MFT) will not be not much different than any other IT infrastructure: plan, scope and delivery incrementally.

Oracle Managed File Transfer (Oracle MFT) is a new product, released with Oracle SOA Suite 12c, which enables secure file exchange and management between internal departments and external partners. It protects against inadvertent access to unsecured files at every step in the end-to-end transfer of files. It is easy to use especially for non-technical staff so you can leverage more resources to manage the transfer of files. The extensive reporting capabilities allow you to get a quick status of a file transfer and resubmit it as required. You can protect data in your DMZ by using the SSH/FTP reverse proxy.

For most organizations, MFT is not new but there is a need for a single enterprise wide file transfer solution that scales to partners and the cloud. Changes in security and compliance requirements are driving IT organizations to replace aging or custom script based file transfer technologies. Encryption such as PGP and Secure FTP servers are still the norm but they must utilize existing Identity Management to define users, groups and roles for remote access. Finally, the need to integrate MFT with existing integration tooling cannot be overstated. Since MFT typically does not support document transformations and other complex orchestration use cases, it is critical for MFT to act as a file gateway when used in tandem you're your existing integration platform. More information on Oracle Managed File Transfer is available in separate documents.

B2B

Oracle B2B 12c is tightly integrated with Oracle SOA Suite and Oracle Managed File Transfer. B2B users will be able to send and receive messages in Oracle B2B by using Managed File Transfer. The connectivity through Managed File Transfer is supported in both directions, towards trading partners and towards the backend middleware. In Oracle B2B, users are able to select Managed File Transfer as a new option while creating delivery channels and listening channels.

Figure 55 and Figure 56 show how Managed File Transfer can be configured as a new option while creating these channels in Oracle B2B.
On the Managed File Transfer side, in order to send a message to B2B, a B2B domain can be created which can be used to send messages to B2B from MFT.

With this integration between Oracle B2B and Oracle Managed File Transfer, a lot of customer use cases can be addressed because many B2B customers have a Managed File Transfer requirement and vice versa.

Oracle B2B 12c also introduces document streaming for large payloads. Many customers had asked for this capability in order to process very large files. Since large files can now be processed as streams, this feature will be very valuable for high volume customers processing large payloads. For enhanced management and monitoring B2B is also integrated with the SOA Error Hospital. It is very simple to track messages end to end by adding direct links to Enterprise Manager Fusion Middleware Control from B2B Reports and vice versa. Support for Local Policy Attachment for Web Services security configuration have also been added, which will simplify the user experience for customers using Web Services in B2B.

Oracle SOA Suite for healthcare integration

The Oracle SOA Suite for healthcare integration is designed to provide healthcare organizations with comprehensive integration capabilities within a unified middleware platform. It helps to reliably exchange information while adhering to important industry standards and initiatives. Oracle SOA Suite for healthcare integration 12c comes with new features specifically for providers including new wizard based installer, enhanced web-services support, support for large documents, Oracle Managed File Transfer (MFT) integration and enhanced end-to-end monitoring.

Wizard based Installer: Oracle SOA Suite for healthcare integration 12c comes with wizard based installer with a new ‘Oracle SOA Suite for healthcare integration template’. The template ensures that all healthcare components are installed and configured. A Healthcare Integration Repository Creation Utility (RCU) is added for the creation of healthcare tables and set up of materialized views in the database schema.
**Enhanced Web-services-Support:** Oracle SOA Suite for healthcare integration allows you to exchange web service (SOAP) based messages between endpoints. You can exchange messages in both inbound and outbound direction. The web service feature not only enables endpoints to receive or send messages, but it is also layered as a protocol implementation and supports other general features such as reporting, tracking, and auditing. In Oracle SOA Suite for healthcare integration 12c, the web-services support has been enhanced even further. You can now attach OWSM policies from the endpoint screen.

![Policy Configuration](image)

Figure 57: OWSM policies can be associated with web service endpoints using the Policy Configuration wizard

**Managed File Transfer (MFT) Support:** Oracle SOA Suite for healthcare integration recognizes Oracle Managed File Transfer as a Remote Endpoint. MFT is added as additional transport protocol in the endpoint configuration: outbound endpoints are used to send files to Oracle Managed File Transfer and inbound endpoints to receive files.

**End to End Monitoring:** The Reports page of the Oracle SOA Suite for healthcare integration user interface (see Figure 58) lets you view the status of the messages being processed through Oracle Healthcare components in real-time. You can view all messages or you can narrow down the messages displayed by a variety of criteria, including endpoints, date ranges, document information, and payload fields. The Reports page is further enhanced to provide end-to-end monitoring for healthcare transactions providing a capability to associate the source endpoint to target endpoints. This is very useful specifically in fan-out scenarios where one source endpoint is sending data to multiple target endpoints.
Figure 58: The Oracle SOA Suite for healthcare integration report page shows end to end monitoring for a fan-out scenario showing association of single source endpoint to multiple target endpoints.
Summary

Oracle SOA Suite 12c and its associated products, Oracle Managed File Transfer, Oracle Cloud and Application Adapters, B2B and healthcare integration, offer the industry's most highly integrated platform for solving the rapidly increasing challenges of integrating with cloud applications, mobile applications, and the Internet of Things. Rather than introducing new toolsets for each new integration challenge, Oracle SOA Suite 12c expands on the proven principles, practices and tools of on-premise application integration to embrace this new world. Enterprises who choose Oracle SOA Suite 12c to connect with SaaS providers, business partners, and the explosion of new automated sensors, will avoid the “accidental architecture” of separate proprietary “silos” and will deliver business-critical solutions faster, with less effort, and with greatly reduced long-term maintenance costs.

Author: Simone Geib
Contributing Authors: Dave Berry, Peter Belknap, Krishnaprem Bhatia, Agnes Freese, Ram Menon, Bruce Tierney, Suresh Sharma, Robin Smith, Robert Wunderlich

Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries
Phone: +1.650.506.7000
Fax: +1.650.506.7200

Copyright © 2014, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 0714