Reactor 5 – Frequently Asked Questions

Contents:

Interfacing with the Server

1. ***What interfaces/APIs does Reactor Server provide?***

   Reactor Server provides several interfaces:

   EJB (using a stateless session bean)
   XML/HTTP
   Java client API (which can use either EJB Session Bean or XML/HTTP)
   SOAP/HTTP (uses XML over HTTP)
   WSDL (uses SOAP/HTTP)

   The EJB interface offers the best performance over a LAN. XML over HTTP can be used by languages other than Java, and is the easiest to tunnel through firewalls. The Java client API provides the most convenience to Java developers.

2. ***How do Reactor clients, such as Reactor Studio and Reactor Portal, communicate to Reactor Server?***

   Any client, including Studio, Portal, and any other client, can send requests to Reactor Server through any of the interfaces above.

3. ***Is the API Java compatible?***

   Yes.

4. ***Can XML be used for interfacing with Reactor?***

   Yes. Reactor Server provides an XML/HTTP interface, as well as SOAP/HTTP and WSDL interface. The XML is described by both a DTD and an XML Schema. The Reactor Server has an internal service that can perform XSL translations on XML requests and responses.

5. ***What's your recommendation for Reactor 5 customers who use Microsoft IIS and ASP pages? How should these pages access Reactor 5 server? Via XML/HTTP client interface?***

   Yes, the XML/HTTP interface would work well. Also, it may be possible to call Java code from ASP pages, which would allow use of the more convenient Java API instead.

6. ***Is there an API that I use to determine what activities a user has to complete?***

   Yes. To programmatically determine what activities are assigned to a particular user, one could send to the server a Query request including an ACL in the query criteria.

7. ***Is there an API that I use to mark an activity as "finished" to move the execution of a workflow forward?***

   An activity can be explicitly finished with the Stop request, or implicitly finished with an AddStatus request for an activity that has a change condition to finish when a certain status is added. These changes, in turn, can trigger preconditions of other activities, thereby advancing the workflow.

8. ***Is there an API that I use to get notification that a workflow execution has been completed?***

   When a workflow Process changes state or status, Reactor fires an event that can trigger Policies to execute. So to send a notification, you would add a Policy that is triggered by the completion of a Process. That Policy would then need to include code to send whatever notification is desired.

Interfacing with External Systems

9. ***How does the Reactor engine handle invocation of external activities that might be involved in distributed enterprise to enterprise business processes, such as Web Services?***

   Reactor 5 Server provides interfaces for EJB, XML/HTTP, and SOAP messages. A JMS interface would be straightforward to add. There is a Java API that provides many convenience methods. This makes it easy to talk to Reactor 5 Server from an external system. To talk to external systems from Reactor, executable policies triggered by process events in the Reactor engine can be written in Java and other languages.

10. ***Does the workflow system have an open architecture that allows systems to respond to workflow events?***

    Yes. For example, an audit trail is implemented by monitoring workflow events. Other functionality could be developed using the same approach.

11. ***Does Reactor support integration with an external calendar?***

    Reactor 5 Policies can be used to check an external calendar service (such as MS Exchange) in order to assign work or make decisions based on the calendar. Reactor does not manage any internal calendar.

12. ***Can Process flows interact with JMS, JCA Connectors, or other back-end systems?***

    Yes. Such interactions would be handled via Policies.

13. ***Can a workflow process be started by an external event, such as a trigger in a database?***

    Yes. Reactor policies can bet set to listen for a variety of events and to perform arbitrary actions – including creating a new process instance – when those events occur. One could also program a database trigger to send a command to create a new instance via any of Reactor 5's supported APIs.

14. ***Can external programs be executed/started from within a process? If so can they be executed/started on both server and client?***

    Yes, Reactor 5 policies can be set to execute external programs when a process is started, stopped, changes status, or when a timer expires. Starting external programs on a remote system is possible via EJB, JMS, SOAP, or a custom call to some sort of agent on the remote system (e.g. 'rsh')

15. ***How does Reactor integrate with Web Services?***

    Using SOAP, with the WSDL basically just wrapping the XML messages as strings.

Other XML Workflow Descriptions

16. ***Is WfMC's standard followed?***

    The Reactor process model was influenced by the WfMC, but it is not followed as a standard.

17. ***Can an external modeller be used easily?***

    The workflow engine provides an XML/HTTP interface, in addition to Java interfaces. External modellers can use this interface to define processes. One could also use an XSL translation layer to accept XML definitions other than Reactor XML.

18. ***Are there plans to possibly support BPML? What is the time frame for that support?***

    Yes, but there is no time frame. This work could be done under contract.

19. ***How well does your proprietary XML schema align with BPML?***

    Fairly close as is, and an XSL translation can be used to bring it into exact alignment.

20. ***How hard would it be to implement full support for standards like WFSL and BPML?***

    There's an internal services inside the engine that can perform XSL translation on incoming XML requests, with corresponding translations on outgoing responses. It would be a matter of defining the translations between the standard of choice and the Reactor XML representation of processes. The Reactor 5 process model is deliberately general. Keep in mind that the Studio only uses a subset of the capabilities of the process model, so don't look there for limitations on the process model itself.

Workflow Capabilities

21. ***Do Reactor support the use of variables inside processes? If so, how are they set, and are they local or global variables, or both?***

    Reactor 5 "Operands" function as variables within a workflow. They can be set from inside the engine, or from outside the engine via API calls. They can be configured individually to have local or global scope.

22. ***Does the Operand class accept only String objects as values?***

    Yes. Operands are intended to work in conjunction with other data storage systems (databases, document repositories, etc.) and not necessarily replace them. That said, objects can be represented as strings using either XML representations or Java serialization.

23. ***Does Reactor support timed activities?***

    Yes, Reactor 5 timers may be set for any activity, along with a Policy to be executed when the timer expires. Policies may perform arbitrary action, such as escalation, reassignment, or reminder.

24. ***If tasks are delayed, can measures automatically be taken (e.g. e-mail notification)?***

    Yes. Reactor supports timers, which can trigger arbitrary actions upon timeout, such as notification.

25. ***Can processes be versioned? Do you have any kind of version control for processes? If so, how?***

    There is no explicit support for versioning processes in Reactor. In Reactor, changing a Process definition has no effect on running Process instances. That is, a Process instance will continue to behave as defined by the previous version of the Process definition. Subsequent instantiations of the Process will use the new version of the Process definition.

26. ***What happens to existing instances of a workflow when a template is updated? Are they finished using the old one, the new one or is the behaviour undefined?***

    Each time a process in instantiated, Reactor creates a complete copy of the process template for use by the new instance. If the template is later modified, existing instances are unaffected. If the instance is modified, the template and other existing instances are unaffected.

27. ***Do you support attachments to activities?***

    Yes.

28. ***What functions do decisions support? (AND, OR, less than, greater than, equal, math, etc.)***

    Reactor's process model does not include an explicit "decision" node. Instead, process activations are determined by pre-conditions associated with each activity. Pre-conditions are defined in terms of Boolean (AND/OR/NOT) combinations of system and external events and "outcomes" of other activities. Scalar comparisons and other expressions must be carried out within the context of an activity and used to determine the outcome of that activity.

29. ***Does the Studio graphical modelling client use AND and OR nodes in the Process diagrams?***

    Yes. These nodes are used to automatically generate the appropriate preconditions for the activities.

30. Can actions be performed on certain conditions?

Yes. Execution of "Policy" scripts is triggered by one of the following events: process started, process finished, status added, status removed, timer expired. Complex conditions can be checked by policy scripts to see whether further actions should be performed.

31. Can subprocesses (nested activities) be created?

Yes. Reactor Processes may be nested to any depth.

32. What automated actions can be associated with tasks and performed by the server?

Any action that can be programmed using Java code or a language supported by BSF (Bean Scripting Framework) can be associated with a task and executed by the server.

Users, Roles, and Security

33. Does Reactor integrate with my directory service (e.g. LDAP)?

Yes. Reactor 5 accesses user, title, and group definitions through a JNDI interface, which can be configured to resolve to any external source, such as an LDAP server.

34. Can roles as well as persons be assigned to a task?

Yes. Reactor 5 activity "participants" can be defined as literals or as "titles" that can be looked up from an external source, such as LDAP.

35. Can tasks be routed when persons/roles are unavailable?

Reactor does not have an intrinsic ability to determine when users/roles are available. However, a Policy associated with process activation could be used to consult an external calendar, Free/Busy, or virtual In/Out board to determine availability of individual users and select among eligible participants.

36. Where is the security information (ACL/UserID/Password) stored?

ACL information is stored in the database where Reactor 5 Server keeps all the other process data. The UserID/Password are validated through a JAAS authentication module, so they may be stored in different places for each module.

37. Is there security on modelling, monitoring, etc.? If so, to what level?

Requests to the Reactor Server must include a valid authentication token, except for the login request that is used to get a token. Each primary type in the process model has an associated access control list (ACL). These ACLs are currently advisory. The original design provides for an internal "Security Service" to enforce authorization using these ACLs. For now, security must be enforced by the client application.

Installation and Administration

38. Which Java virtual machines are supported/required? If any.

A JVM with version 1.3.1 or higher is required.

39. What EJB/Servlet/JSP versions are supported?

Reactor supports the J2EE 1.3 versions of the EJB/Servlet/JSP specs. With some additional steps at deploy-time, Reactor's descriptors can be modified to work with the J2EE 1.2 versions of these specs.

40. How does Reactor access the database to store workflow data?

The server uses JDBC to access a J2EE data source provided by the application server. The administrator of the application server configures the data source to connect to the database desired.

41. Which platforms do you support?

The Reactor 5 Server runs on any J2EE 1.3-compliant application server, and has been tested on the following application servers:

+ *JBoss with Tomcat*
+ *BEA WebLogic*
+ *WebSphere*
+ *Oracle 9iAS Application Server*
+ *Orion Application Server*

The following databases have been tested successfully:

+ *Oracle*
+ *SQL Server*
+ *PostgreSQL*
+ *InstantDB*

Reactor 5 Studio runs on any client platform that supports JDK 1.3.1 or higher.

All other Reactor 5 clients are completely browser-based, and will function with any HTML browser.

42. Why does Reactor5 require an application server? How tight is the dependency or integration?

Reactor 5 Server uses a collection of EJB stateless session beans to implement internal services. Removing Reactor 5 Server from the J2EE application server would require replacing the code that does service lookup and request/response handling. The architecture supports this, but Reactor 5 has been assuming a J2EE platform for some time now, so the implementation may have that assumption in unexpected places by now. Reactor 5 Server does not use container managed persistence, but does use the application server's data sources for configuring and pooling connections.

Reactor 5 Server uses servlets to provide XML/HTTP and SOAP interfaces to external clients. It provides an EJB stateless session bean as another interface for external clients. The Reactor Portal also uses JSP and servlets to build web-based workflow applications. Timers are managed by a servlet that sends scheduled event notifications. This requires an application server that at least supports servlets and JSPs.

43. Are you able to operate in a load-balanced environment and still maintain proper state as events end up on different servers in a cluster?

Yes.

44. What third party software does Reactor use/require?

Reactor Portal uses the com.oreilly.servlet package, which has a license requiring each ***developer*** *to have a copy of the O'Reilly servlet book, but has no distribution restrictions.*

Versions prior to Reactor 5.0.4 used Flux for scheduling, but that has replaced with custom code that does not require a license.

All the other required third party software is open source with no restrictions.

Ant to build from source
SOAP for web services
Struts for JSP tags
Xalan for XSL translation
Xerces for XML parsing
BSF for scripting
JDOM for XML representation
JUnit for unit testing

45. Does Reactor use container-managed transactions or explicit transaction demarcation for the EJBs?

Explicit transaction demarcation. The current level of transaction support is that each individual request has an internal two-phase transaction. This transaction does not currently participate in the scope of a larger client transaction.

46. I'm getting an OutOfMemoryError. What should I do?

Increase the memory allocation to the JVM. Edit the start script for your application server. Use the –xmx and –xms options of the java executable to increase its memory allocation.

47. Why do I get an error message when I try to point my browser at the "Front Desk" servlet?

The Front Desk servlet is not intended to be accessed directly by web browsers. Client code should POST requests to this servlet for processing by the server. If you are evaluating Reactor, use the Reactor Portal client application, which is usually installed at the context root /ReactorPortal/ in your application server.

Localization

48. Does the engine support DBCS and Unicode?

Reactor depends on the JVM, application server, and database engine to provide transparent handling of character sets. Reactor does not support or depend on any particular character set.

49. Does the user interface support localization?

Yes. The Reactor 5 Studio uses resource bundles, although there are some places where it should use them where it doesn't. The Reactor 5 Portal can use resource bundles, since it uses the Apache Struts framework, but replacing each entire JSP pages might also be an option.

50. Does the workflow engine support localization of internal strings?

The workflow engine currently has strings embedded in the code. In general, the strings handled by the workflow engine are only visible to the user if they are success or failure messages.

51. Does the workflow engine support localization of workflow data?

No.

Reactor Portal

52. How and when are new items in the work list updated?

The Reactor 5 Portal Framework is a web-based client that can be used to view a user's work list. The work list is regenerated from the Reactor 5 database each time the user opens or refreshes their work list URL. The Reactor 5 database is updated with every transaction. An HTTP Meta tag could be used to achieve periodic client-side refreshing.

53. Does your work list support filtering of tasks?

This would require customization of the Reactor 5 Portal, which can be done by modifying Java Server Pages (JSP).

54. Is there a monitoring tool?

Reactor 5 Portal includes some monitoring capabilities.

The Reactor Source Code

55. What principles and/or domain knowledge guided Reactor's design?

Reactor was specifically designed to be flexible and not domain-specific.

Reactor's architectural structure is internal services which are Stateless Session Enterprise Java Beans (EJBs), with "Front Desks" that provide the client interfaces. The R5 service architecture was heavily influenced by Jini.

Enterprise layers use J2EE where available, modular interfaces otherwise. (ex: portal has document repository module)

Process model started with R4 and was redesigned to be more flexible.

The Reactor development team adapted XP for its development methodology, which influenced Reactor's design.

56. *In what language is the engine implemented?*

100% pure Java.

57. *What is the Size of the reactor code? (breakdown into Studio/Server/Portal)? * Any non-java/non-script code? (native/C?)*

Code is all pure Java, with simple launch scripts in sh and .BAT files. Scanning the current version (with some development in progress), reveals the following:

Server: [433 files]
7671 lines with just a bracket (12.99%)
7135 lines with just whitespace (12.08%)
9181 lines with just comment text (15.54%)
35077 lines with 'significant' code (59.39%)
59064 lines total
Studio: [72 files]
2296 lines with just a bracket (13.82%)
2215 lines with just whitespace (13.33%)
1389 lines with just comment text (8.36%)
10719 lines with 'significant' code (64.50%)
16619 lines total
Portal Framework: [31 files]
546 lines with just a bracket (14.99%)
537 lines with just whitespace (14.74%)
775 lines with just comment text (21.27%)
1785 lines with 'significant' code (49.00%)
3643 lines total
Active Checklist: [11 files]
78 lines with just a bracket (8.70%)
127 lines with just whitespace (14.16%)
310 lines with just comment text (34.56%)
382 lines with 'significant' code (42.59%)
897 lines total

There are also some JSP files with a total of less than 1000 lines.

58. *How is the engine delivered, e.g., source, jars, etc.*

At present, source, including all documentation, CVS repository, test scripts, ... our complete development environment.

Miscellany

59. *Are statistics saved? Can statistics be viewed? If so, how?*

Reactor 5 stores a complete trace of the execution of every process instance. Reactor does not pre-generate statistical summaries of this data, though statistics can be generated using a suitable reporting or OLAP tool. Reactor 5 does not include a statistics viewing client.

60. *Can I record an audit trail of Process events?*

Yes, the reactor distribution includes several Policies that record an audit trail to an arbitrary database. It can be configured to store different types of events.

61. *I'm getting an error that says something about java.lang.Process. What is this?*

*When writing java code that uses the com.oakgrovesystems.processMediation.Process class, one must explicitly import this class by typing **import com.oakgrovesystems.processMediation.Process;** in addition to **import com.oakgrovesystems.processMediation.*;.** Otherwise, the compiler will not be able to determine whether subsequent references to the class "Process" refer to com.oakgrovesystems.processMediation.Process or java.lang.Process.*

Last Updated: February 3, 2003