

# RBA best practices

The following best practices provide suggestions for how to successfully work with RBA.

- **Decide what to automate**—Do an audit of what happens with a typical job with all the operator interventions and interactions, and identify the points where automation might add value. Likely candidates are operator actions where operators "wait until X, and then start Y," or "do X, and then Y, and then Z." Automated tasks should be repeatable and predictable, with a similar workflow each time.
- **Brainstorm**—Draw the rule set on paper or on a whiteboard before you start work in Rule Set Builder. Have a clear picture in your head about the flow of the rule set, what parameter values you'll need, and what you expect out of each action.
- **Plan ahead**—Try to imagine all possible inputs to your rule set and then accommodate them. What effect (if any) will this rule set have on other rule sets in the same or in system context? What will happen if this rule starts unexpectedly? What could happen if the rule set acts on wrong information at runtime? What will happen if this rule set is started hundreds or thousands of times?
- **Contingency plan**—How do I stop this rule set if I have to?
- **Debugger**—Build rule sets progressively, one rule at a time, validating behavior in the debugger before extending the rule set. If you construct a rule set in this way, you allow the system to help you by providing assignment hints for parameter values where possible.
- **Develop**—Use the manual trigger as well as the usual trigger event of the rule set while building and testing it. This way, you can trigger the rule set whenever you need.
- **Audit**—Every rule set should have a rule chain, such as **Exception Handler > Log To History**. This is a generic exception handler that will capture valuable information about any rule exception and log it to history.
- **Naming**—Give a meaningful name to any significant actions so that the observer can glean the rule set writer's intention. Rename (resulting) events and give meaningful (short) names that will, in turn, appear in the Event Properties browser list.
- **Efficient rule sets**—Build for code efficiency. That is, build your rule with a view to minimizing the number of steps—for example, use a table of values, rather than a monster chain of branches.
- **Document**—Use the **Rule Set Description** box to give an appropriate description of the intended purpose, the prerequisites, and the expected outcomes of the rule set.
- **Encapsulate**—Identify external resource dependencies that may change in the future and try to encapsulate them in global or rule set variables. Some examples are e-mail addresses, server names, files shares. A global variable should be used if the value is going to be referenced across multiple rule sets. A rule set variable should be used if it is only going to be referenced from that rule set. By encapsulating them in a variable it will be easier to update them in the future should the resource names change.