InRule Technology®
Getting Started Guide

inrule
TECHNOLOGY®

Welcome to the InRule Technology Getting Started Guide

At InRule, we understand and automate decisions. We do it faster, with more accuracy and transparency than anyone else.

InRule® empowers technical and business users to change rules and calculations in applications—with less effort, cost, and risk than programming—to help organizations keep pace with changing marketplace conditions and customer needs.

Determining if a software product is the right solution for your organization can be a time-consuming and sometimes confusing endeavor.

To help you with your evaluation and provide guidance on what you need to look for, and be aware of, during that process, the InRule Technology Evaluation Guide will walk you through key aspects of our business rule management system.

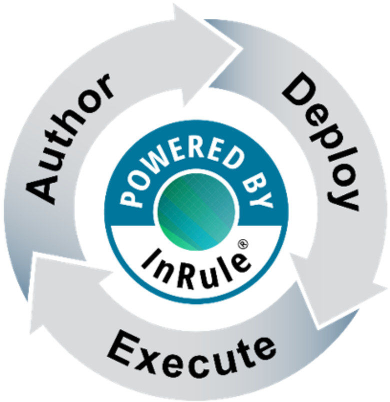
We've made some assumptions in developing this guide. You are well aware of what a business rule is and the difficulties, challenges or complexities that arise when business rules are hard-coded into an application. What *you* need to know is how InRule can help.

At the end of this guide you will find a list of potential next steps for your evaluation process. One of our sales representatives will contact you to discuss which step or steps you prefer to take during your evaluation phase.

Table of Contents

Introduction

What is a business rule management system?



A business rules management system, or BRMS, is an end-to-end solution for everything related to a business rules lifecycle, from authoring to testing to storage and execution.

The InRule BRMS is industry-agnostic. Because every organization's rules are different, we provide a solution that can fit into *any* business application. From insurance, healthcare, financial services and other commercial applications to public sector agencies, InRule makes it easy to author and manage rules.

While our BRMS is built in .NET, you don't have to be a .NET organization to use InRule. With today's service-based architectures, InRule integrates with almost any type of application (desktop, web, mobile) or platform (Microsoft Dynamics®, Salesforce®).

A BRMS can be your "Single Source of Truth." Instead of replicating logic for each of your business applications, define that logic once and let all of your applications consume the same rules consistently.

So what's a business rules engine?

A business rules engine, or BRE, is the algorithm that executes the business rules and is just one component of a BRMS.

Components

How do I write my business rules?

irAuthor® is the first component of the InRule BRMS. It is the desktop tool you will use to write and maintain your business rules using one or more of the included rule editors.

The Business Language rule editor provides a point-and-click interface for constructing rules using out-of-the-box or custom templates in order to implement logic in a natural language representation.

The Decision Table editor can be used to represent a set of related business rules in a multi-dimensional matrix by combining multiple conditions and actions.

The Syntax Expression Designer allows rule authors to build complex expressions or formulas using any of the 225+ built-in functions.

Use these rule editors to create the business rules that represent a use case or decision. Together, this collection of rules is referred to as a rule application.

What about testing business rules?

An essential step in the business rule lifecycle is testing. irVerify® is the business rule testing tool. With irVerify you can execute unit tests or regression tests. You can also view logging information captured by the rule engine during each test execution so you can review individual rule results, state changes or final decision results.

irAuthor

Components

Where do I store my rule applications?

irCatalog® is the second component of the InRule BRMS. After you've created a rule application using irAuthor, you will save it to irCatalog.

irCatalog is source control for your rule applications. It consists of two parts: a database and a web service. The database will be the repository and the web service will be the mechanism used in irAuthor or through programming interfaces to communicate with the database and retrieve a rule application.

After a rule application has been retrieved from irCatalog and opened in irAuthor, you will need to check out the rule application to make your rule changes. Once you complete your modifications you will check them back into irCatalog.

This change process creates an audit record in the irCatalog database. You can view the history of a rule application to view who made changes to it and when.

Is there security built into irCatalog?

irCatalog includes out-of-the-box roles and permissions, but most of our customers use Active Directory to leverage their Users and Groups in irCatalog. Select a User or Group and assign them rule permissions. They can be administrators, editors or just viewers of rule applications. You can also create your own permissions.

If you aren't using Active Directory, any LDAP compliant system will work in its place.

irCatalog

Components

How does a rule application get executed?

irServer® is the name for our BRE. Pass in data (such as a record or a transaction) and a rule application, and irServer will read the rule application to determine which rules it should execute and which rule it should start with. When it is done executing the rules, it will hand back the resulting modified data.

Rule execution with irServer is stateless. That means we never store the record data that is passed in. irServer operates on the data and removes it from memory when it is finished.

In the following pages, we'll dive into the common ways you can integrate a call to irServer from a business application.

Is your rule engine RETE?

During the course of your research into rule engines, you may have encountered this term. RETE is the original business rule engine algorithm created way back in 1974. Some of our competitors may implement a RETE algorithm, but we do not. Why? We decided to create a proprietary algorithm. irServer's algorithm is flexible enough to handle different execution modes depending on how you design your rule applications using the rule application configuration settings in irAuthor.

irServer

Components

How do I call irServer?

Option 1: irServer Rule Execution Service

Your first option for integration of business rules into your application is to use InRule's out-of-the-box irServer Rule Execution Service.

We provide two implementations in the same service, SOAP or REST, depending on your organizational or architectural preference. Using this service provides the added benefit of a quicker deployment since it requires no additional coding because we've already done that work for you. Simply deploy it and it's ready to go.

Can we use this in the Cloud?

If you are looking to go Platform as a Service (PaaS), the irServer Rule Execution Service can be deployed as an Microsoft Azure® Classic Cloud Service using our pre-built Azure package and configuration files. If your preference is Infrastructure as a Service (IaaS), whether your organization uses Azure or AWS, you can run the InRule installation on a Cloud VM to install the irServer Rule Execution Service.

If you deploy rule execution services to the cloud, you will also deploy irCatalog, its service *and* the database, to the cloud as well.

Components

How do I call irServer?

Option 2: Build custom services using irSDK®

If your organization has specific requirements for how your services must be implemented, perhaps due to security guidelines or data contracts, then your next option is to use irSDK® to create your own custom rule execution service(s).

This is the path most of our customers take. Our out-of-the-box service, which was actually created by InRule engineers using irSDK, is great for simple request and response interactions, but when you need greater control or flexibility with the implementation *and* you have the available .NET programmers, then this is the way to go.

Using irSDK you can create generic rule execution services that are parameterized to your needs or individual REST operations. You can deploy your custom service on-premises or in the cloud.

Maybe you need to execute business rules in-process in a .NET application, whether it's WinForms, WPF or ASP.NET. If that case, just add the InRule assembly references to your project and call irServer directly through the programming interfaces.

No .NET programmers? No problem. We have a professional services team that can work with you to get that integration exactly as you need it.

irSDK

How do I call irServer?

Option 3: Package rule applications as executable JavaScript

If your application requires client-side rule execution, disconnected rule execution, server-side rule execution via Node.js, or you’re just a JavaScript shop, we provide InRule *for JavaScript*.

You will still author, test and store your rule applications using the same tools, but when you want to take a rule application and repackage it as executable JavaScript, you simply upload it the InRule-hosted irDistribution^{TM} service. irDistribution will convert your rule application and send back a JavaScript file. This file will include everything wrapped in one: the business rules, application programming interfaces and irServer.

Your developers can copy the JavaScript file into the code project and use the embedded APIs in that file to execute the business rules, regardless of whether you are building a custom application or you are using a JavaScript framework like Angular, Knockout or ReactJS.

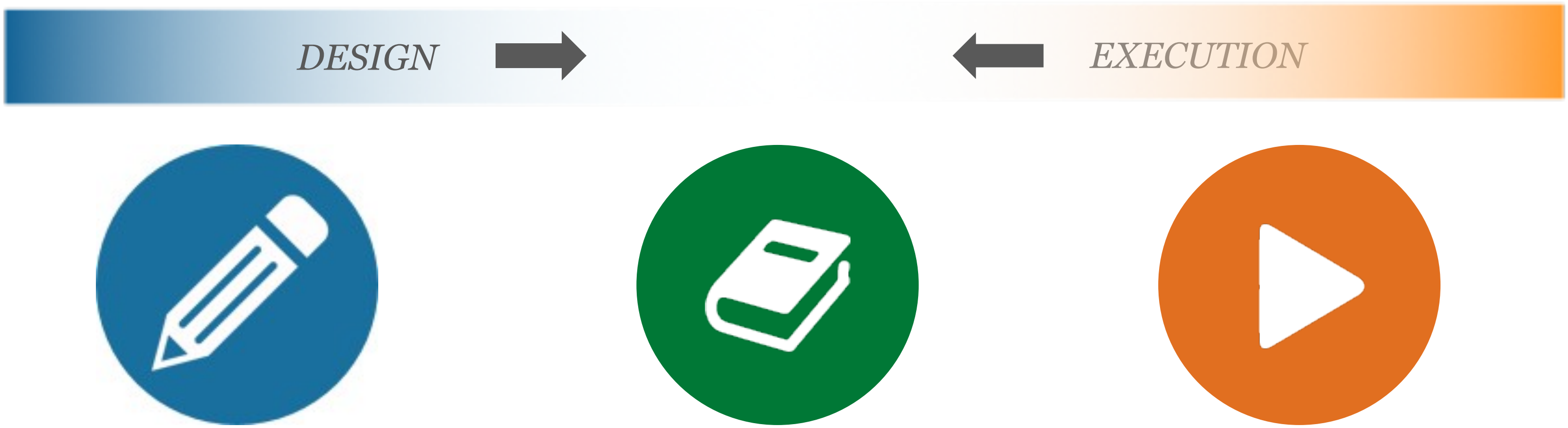
Can we do serverless JavaScript execution?

If you like the idea of having executable JavaScript rules, but need the flexibility of serverless execution, simply deploy your JavaScript rules as an AWS Lambda Function or as an Azure Function.

irDistribution

Components

How do the components connect to each other?



DESIGN →

← *EXECUTION*

irAuthor

irCatalog

irServer

The design phase entails writing the business rules using irAuthor, testing them with irVerify, and storing those business rules in irCatalog.

irCatalog is the link between the design and execution phases of the business rule lifecycle and represents your “source of truth.”

The execution phase entails the rule engine pulling the rule application from irCatalog.

Architecture

BRMS Component Connections

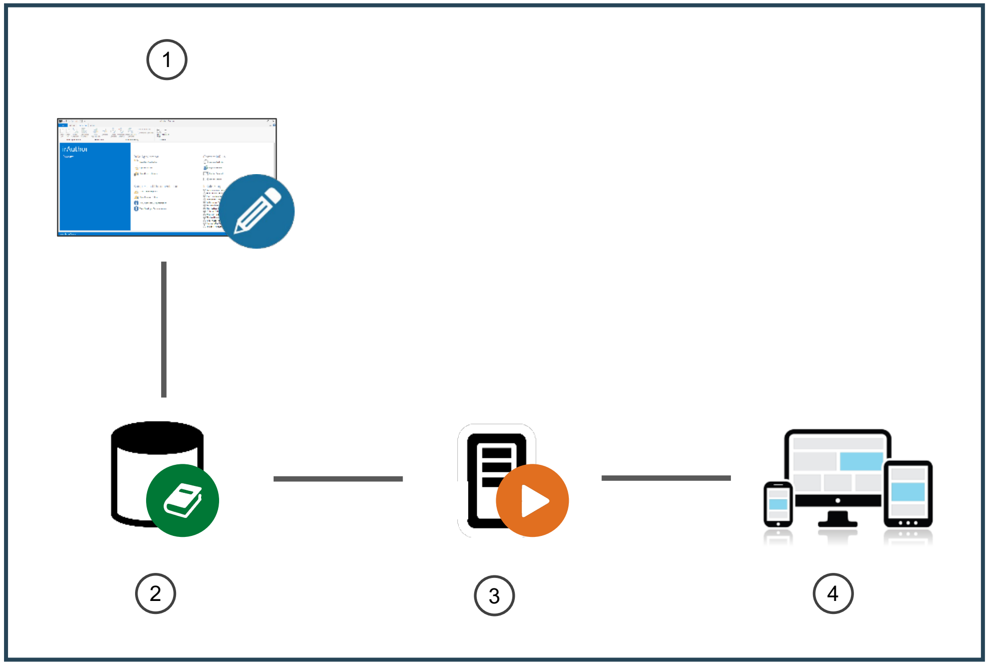
The most common architecture is a service-based integration. We'll start with a generic diagram to cover the connection points.

Step 1 is to use irAuthor to write the rule application.

Step 2 is to save the rule application to irCatalog.

Step 3 is to deploy a rule execution service and configure it to point to an irCatalog instance.

Step 4 is to configure your business application to call the rule execution service.



This diagram represents a single environment.

Architecture

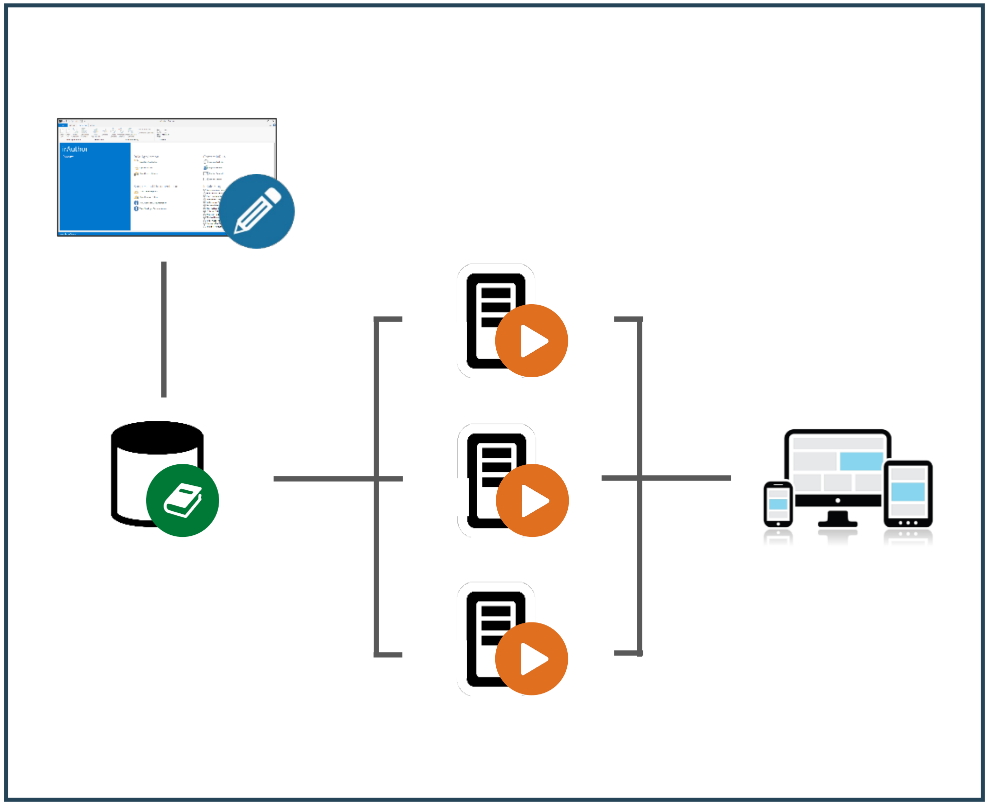
Scaling a Rule Execution Service

When your business application needs to process larges volumes of records or transactions through the rule engine, the rule execution service can be scaled out accordingly.

For a single environment, all of the scaled rule execution services are configured to point to the same irCatalog instance which will act as your source of truth.

Does your BRMS include a load balancer?

No, but when you need to distribute transactions across rule execution services, you can use your existing or preferred load balancer. This provides a familiar and consistent distribution model within your current service architecture.



This diagram represents a single environment with scaled rule execution services.

How fast is the rule execution?

We are often asked how fast the execution will be. Unfortunately, it is not feasible for us to state a specific number since every customer implementation varies.

In our world, a long-running rule execution would be anything over 150ms per transaction. Most of our customers are executing rule applications in much less than that time, and some as low as single digit milliseconds.

What we *can* tell you is that there are two primary factors that will contribute to your business rule execution times:

Factor #1: Number of rules

It may seem obvious, but it's worth stating – the more rules in your rule application, the longer the execution can take.

Some rule applications may only have a handful of rules. Some rule applications may have thousands of rules. What is important to know is that not every rule will execute every time because conditional statements and branches will direct the rule execution sequence.

If large rule applications are unavoidable, our services team can provide best practice guidance on proper rule application design to optimize performance.

Factor #2: Size of record data

When the data is passed in to the rule engine, irServer must map it into a corresponding data structure. After the execution is complete, it must then map the results back.

Some rule applications may call for passing in minimal record data since the rules are designed to operate only over a smaller set of fields on the record.

Some rule applications may call for passing in a complex hierarchy of data that consists of parent-child relationships and collections.

How many cores will I need to license for irServer?

We understand how important it is to have the right amount of rule execution power to handle your particular application requirements. Performance is critical. Your application users expect nothing but the best. What's equally important to you is the cost. The combination of these two factors make estimating the appropriate number of cores for irServer an important decision point both for the success of your application and for your project budget.

As mentioned in the Performance section, the magic formula to estimate a transaction time does not exist. The factors, number of rules and data payload, are too variable from customer to customer and even from rule application to rule application.

Instead, we can do some reverse engineering to arrive at an estimate for the necessary number of cores. To make that happen, there are two pieces of information we will need from you:

First, tell us your goal rule execution time for a single transaction. This *should* be in milliseconds.

Second, tell us how many transactions you anticipate executing in a given amount of time. Some applications are running 24x7. Some applications have peak volumes during certain business hours. Determine which is most relevant for your application and estimate on the high end. For example, Company ABC may process loans with their web application. They estimate processing 5,000 loans each day.

When you get to a point where you need a quote, we can use this information, along with the relevant numbers for the other components, to determine the price.

Core estimation does not apply if you determine that your application will *only* run executable JavaScript rules. In this scenario, cores are not a factor since a Windows-based rule execution service is not required.

Licensing

How are the components licensed?

InRule is licensed using a subscription based pricing model with maintenance included.

Now that you have a better understanding how the different components of our BRMS are used, the next question you will no doubt have is how they are licensed. While the entire solution is licensed as a subscription, there are certain components that are configurable. irAuthor and irSDK are based on named users. irServer and irCatalog will depend on what you need in a production environment and therefore will require a discussion with an InRule Sales Engineer to determine the proper sizing.

What's included in maintenance?

Your subscription will include access to the InRule Support Site for forums, videos, and updates, unlimited service and minor point releases, phone and email support, unlimited use of our computer-based training for irAuthor, and major version upgrades.

Can we OEM InRule?

Yes. InRule can be embedded within your product or solution which gives you the option of letting your customers tailor their own business logic in your packaged solution and devices. This can be a key differentiator between your offering and your competitor's.

irAuthor

irCatalog

irSDK

irServer

irDistribution

Roles and Responsibilities

Who does what and when?

The success of your BRMS implementation depends on having the right people doing the right job. Every business rules-based project should consist of the following roles:

Rule Architect

Uses irAuthor to implement the framework of a rule application

Responsible for the import or refresh of the data schema, configuration of connections to services or databases, and creation of custom templates

Will use irAuthor frequently at the beginning of a project to ensure the Rule Authors have everything they need to write rules

Rule Author

Uses irAuthor to fill out the rule application framework by implementing the business logic

Responsible for authoring business rules using Business Language, Decision Tables or Syntax Expressions

Has the ability to interpret use case requirements into business rule constructs

Most frequent user of irAuthor

Rule Integrator

Uses irSDK to integrate the execution of the business rules into an application

Must be able to use application programming interfaces in either C# or JavaScript

Must be familiar with service-oriented architectures

May use irAuthor to assist with more technical aspects of a rule application

Roles and Responsibilities

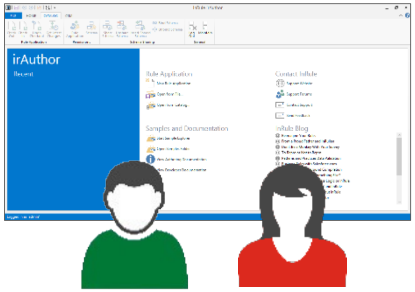
What skills are needed to use irAuthor?

For more than 15 years, we have worked with every one of our customers on their projects. During that time we found there are certain competencies that translate into being an effective business rule author. These are foundational skills that a Rule Architect or Rule Author should have to fully leverage the capabilities in irAuthor to create and maintain rule applications.

Additionally, every irAuthor user will go through a training session conducted by one of our consultants. This session will cover all of the features available in irAuthor and ensure that your rule authors understand the best practices that should be followed to build stable and maintainable rule applications.

The following list represents key rule authoring skills:

- Understand data structures with parent-child relationships
- Translate a use case into a list of distinct, ordered steps
- Combine functions to create formulas or logical expressions
- Identify and implement cause and effect logic using If/Then/Else statements
- Implement table lookups based on matching criteria
- Ability to identify logic gaps



Professional Services

Can we get help from InRule on our project?

Our Professional Services team, known as ROAD (**R**ule **O**riented **A**pplication **D**evelopment), is involved in every one of our customer's projects. ROAD consists of Rule Authoring Consultants and Rule Integration Consultants.

Rule Authoring Consultants will train your designated Rule Architect(s) and Rule Author(s) on irAuthor. They can also perform rule application reviews or provide best practice guidance and samples through the design phase of your project.

Rule Integration Consultants will train your designated Rule Integrator(s) on irSDK. They handle the more technical aspects of the architecture and integration and can provide guidance on deployment practices, performance optimization, capacity planning and state management.

How much ROAD time will I need?

We recommend a minimum of 5 days. The average engagement is 15 – 20 days. During the sales cycle, we can determine if more time is needed based on your project requirements and any time and resource constraints. The time you contract with ROAD can be scheduled in blocks of days or hours, depending on when you need them. And you can always add more ROAD time if you run out.

Core Services

Training	Proofs of Concept Prototypes
Rule Application and Architecture Review	ROAD Sessions Best Practice Guidance

Advanced Services

Rule Harvesting and Authoring Guidelines	Performance Optimization and Capacity Planning
Advanced Deployment and Testing Practices	State Management Data Integration Scenarios

Next Steps

What are my options for moving forward?

Intro Product Demonstration

One of our Sales Engineers will take you through a demonstration of irAuthor, show you how to write and test business rules, and answer any questions you may have as it relates to the product or your project.

Custom Demonstration

When you want to see an example of one of *your* use cases built with irAuthor, send us some documentation on your business rules and some sample data and one of our Sales Engineers will create and present a custom demonstration. This is a great opportunity to get a better understanding of how your rules will look and hear about some of the rule authoring features you could leverage for your project.

Self Evaluation

Visit our Download Request page to access a free, 30 day trial copy of the full InRule BRMS. During the course of your self evaluation, you will have access to one of our Sales Engineers who can provide guidance or answer any questions you may have.

Pilot

If your project depends on small-scale BRMS integration to prove out a long term implementation, you can speak with one of our Sales Representatives to discuss pricing, sizing and professional services options.

Request a Quote

Ask to speak with one of our Sales Representatives to discuss licensing and pricing.

Visit our website to view more information about our product and services, customer stories, partner programs or download white papers.