

# **Better Business Apps with InRule<sup>®</sup>, Microsoft Dynamics 365 and the Power Platform**

---

**inrule**

# Contents

- Introduction. . . . . 3
- Customer Centricity in Patient Care: An Integrative Approach. . . . . 3
- Authoring with irX. . . . . 4
- Authoring Rules . . . . . 4
- Testing Rules. . . . . 7
- Deploying Rules to an Environment . . . . . 8
- InRule and the Power Platform . . . . . 9
- InRule Integration Methods. . . . . 10
- Additional Integration with Power Automate . . . . . 11
- Conclusion . . . . . 12

## Introduction

Nearly 60% of all custom applications are now built outside the IT department, with nearly a third being developed by employees with limited or no technical development skills. Some estimates say that by 2024, low-code application environments will be responsible for more than 65%<sup>1</sup> of app development activity. As digital transformation continues to remove technical barriers and enable greater business agility, putting customer outcomes—not systems, nor data—at the center of application development will be central to gaining a competitive advantage.

Granted, focusing on the customer is not new; traditional customer relationship management (CRM) systems were aspirationally designed to put the customer at the heart of engagement. In fact, the explosive adoption of CRM systems across every industry—including medical, insurance, finance, manufacturing—has been a direct response to the overwhelming majority (84%)<sup>2</sup> of consumers saying they consider their experience of a brand as important as the product or service it provides.

To shape an experience into one that is both personalized yet scalable across an enterprise, on average, organizations are expected to use data from at least 15 different sources. Unfortunately, many CRM systems, rather than providing an integrative experience, inadvertently created systemic and informational silos as companies attempted to adapt them to their business processes.

It was because of challenges like these that the Power Platform emerged.

With Dynamics 365, the Power Platform, and a customer-centric approach, building applications to deliver tailored outcomes in real time becomes a holistic endeavor. Let's look at a series of care coordination scenarios, with a patient as the customer.

## Customer Centricity in Patient Care: An Integrative Approach

The patient has a variety of data sets and communications from their primary care physician, specialists, labs, hospitals, pharmacy, billing, insurance, and so on. Collecting and presenting this data centered around the patient (instead of each individual data channel) ensures that the patient receives the most appropriate care while enabling control and monitoring of that care's cost.

Power Apps, a component of the Power Platform, is a low-code solution that allows developers to quickly create cross-platform applications to extend the use of the data collected and structured by Microsoft Dynamics 365 and the Common Data Service (CDS). This could involve several different functions, such as prescription recommendation management, rebate management, health care team integrations, drug interaction automation, or fraud detection.

Power Automate, another part of the Power Platform, helps automate workflows to make routing information, actions, and decisions to the right place easier. For example, a developer could automate a workflow for suggesting, approving, and scheduling a medical procedure, resulting in faster service and reduced need for manual intervention.

1. Impact Networking, "10 Low-Code Trends for 2020", Impact Networking, <https://www.impactmybiz.com/blog/blog-low-code-trends-2020>, (accessed June 3, 2020).

2. Z. Totah, "7 Critical CRM Trends to Keep an Eye on For 2020", SelectHub, <https://www.selecthub.com/customer-relationship-management/crm-trends/>, (accessed June 3, 2020).

However, in addition to collecting and analyzing data, the best possible care for the patient requires sound decisions made for each “next step,” in alignment with practice policies and regulatory guidelines, and applied consistently and accurately every time. **This is where InRule® comes in.**

InRule enables users to design, implement, and maintain business rules without having to write code. InRule for Dynamics 365 (“irX®”) is a business rules management extension that delivers decisioning capabilities to Dynamics 365 and the Power Platform. With irX, business users can build and test directly with Dynamics 365 data. InRule also supports the integration necessary to execute rules and decisions directly from within these powerful Microsoft platforms.

The suite of tools that includes Microsoft Dynamics 365, Microsoft Power Platform, and InRule’s irX come together to enable automated, yet highly personalized, patient-centric, and in-context decisions—in real time.

This paper explains how InRule, Dynamics 365, and the Power Platform work together to enable consistent design and automation of complex decision-based workflows. There are two primary parts of this integration:

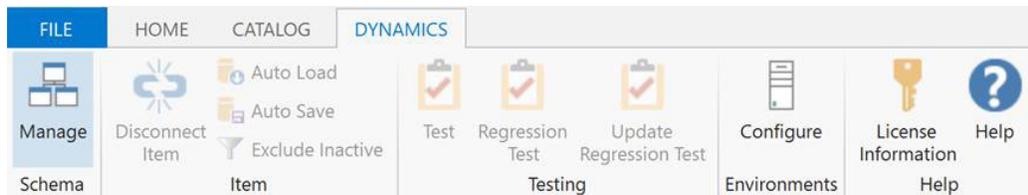
- Authoring and testing rules and decisions
- Integrating rules with the Dynamics 365 and Power Platform environment

## Authoring with irX

irX is an extension to InRule’s popular irAuthor® environment that provides an intuitive experience for seamlessly writing and testing rules with data in Microsoft’s CDS. irAuthor’s user interface (UI) makes it easy for business users to develop and maintain business rules without writing code, no matter how complex or varied the data. In addition, users can configure the irAuthor environment to accommodate both technical and non-technical authors.

Dynamics 365 and the Power Platform leverage CDS, creating a shared backbone for interacting with data. The authorized Dynamics 365 and Power Platform instances have access to all entities, relationships, and data in CDS. With built-in CDS integration, irX enables InRule to seamlessly operate against these same CDS entities, data, and relationships.

irX provides a ribbon toolbar to manage all Dynamics-related rule authoring and testing functions. One of the unique features of InRule is its ability to control the loading and saving of data throughout the entity schema, which greatly simplifies the need to address data management concerns in the rules themselves.



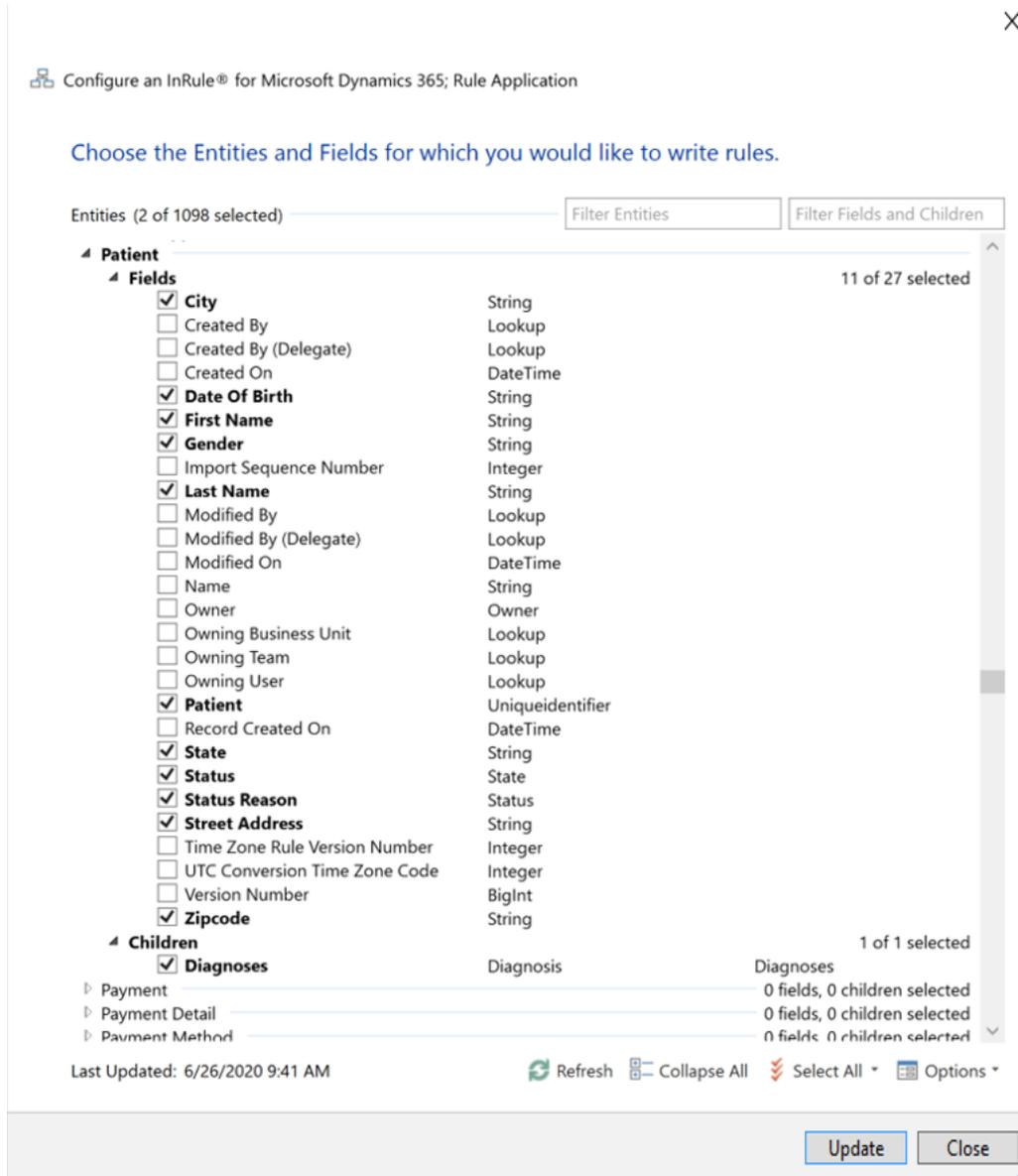
» Figure 1 irX Ribbon Toolbar with Auto Load, Auto Save and Schema Management

## Authoring Rules

Rules are created within a rule application. To begin authoring rules, users simply select the CDS entities and fields they wish to use by selecting them in the connector configurator. This forms

what’s called the “entity schema,” and the fields are mirrored within the rule application using the business-friendly display names from the CDS. Regardless of their complexity (1-to-n, n-to-1, n-to-n), relationships between data are easily handled within the InRule environment. In addition, the majority of CDS data types and option sets are supported, and schema changes can be refreshed and resynced between CDS and InRule at any time.

Going back to the patient example, here might be a connector configuration dialog:



» Figure 2 Connector Configuration Dialog

InRule’s user interface then empowers those who know the business best to automate mission-critical, complex decisions, by authoring rules. Rules are written in plain business language, without the need for complex code.

**If**

- all of the following are false
  - ▶ **Patient Age** is between 18 and 75
  - ▶ any members exist in **Patient Encounters** filtered by all of the following are true
    - ▶ any members exist in **Encounter Diagnoses** filtered by **ICD10 Code** is in **Diabete**
    - ▶ **End Date/Time** is within 3 years of Today

[add condition]  
[add condition]

**Then**

Add referral for "Diabetic Eye Exam" for the patient with a priority of "Routine"  
[add action]

» *Figure 3 Business Language Rule Editor*

Our context-aware, menu-driven editor offers prompts designed for the data being incorporated into the rule, presenting the best possible choices for implementing the unique business decisions to be executed.

**If**

- all of the following are true
  - ▶ **Appointment Status** has a value of "Checked-In", [list value]
  - ▶ **Patient InsuranceType** does not have a value of "PPO", [list value]
  - ▶ **Patient Coverage Verified** is false

[add condition]

**Then**

Notify user that patient insurance is not verified  
Verify [VerificationType] for patient

Value List
Valid Insurance Coverage
Photo ID
Current Address

» *Figure 4 Context-Aware Choices Presented to Users*

In addition, every organization, practice, or department has its own language—a way of speaking about processes and policies that may be wholly unique to them. This is why InRule developed our vocabulary feature. Users can create templates from InRule expressions that can be saved within a single rule application or used across rule applications with a simple copy-paste. This reusable logic, or vocabulary, further enhances ease of use, accelerates the development of decisions, and increases business agility by providing non-developers with tools built in the language of their daily work life.

Lastly, rules can include messages that document why a specific decision was made for traceability, transparency, and auditability.

**If**

all of the following are true

- ▶ **Procedure Type** has a value of "Elective", [list value]
  - ▶ no members exist in **Patient Referral Requests**
  - ▶ **Practitioner Practitioner Type** has a value of "Specialist", [list value]
  - ▶ **PCPOverride** is false
- [add condition]

**Then**

mark **this Appointment (EMR)** invalid and display "Procedure is elective and not authorized by PCP"<sup>^</sup>

[add action]

» *Figure 5 Increase Transparency and Accountability with Messages*

## Testing Rules

Once rules have been authored, users can test them against actual Dynamics 365 data using irVerify®, a part of the irX extension for Dynamics and the Power Platform.

The "Test" command initially prompts the user to select a specific record from Dynamics 365 for the chosen entity context. This record will serve as the primary input data to load along with all nested child collections and related records. The complete dataset will then be used to populate irVerify for testing rules and decisions.

Select Test Data

Use the views from Microsoft Dynamics 365 to locate test data.

Appointment (EMR) view: All Appointments (EMR) Environment: InRule Sandbox

Search Test Data [Search]

Subject	Start time	Patient	Practitioner	Actor (Location)
Bryon Denman - Appointment	10/21/2018 10:36	Bryon Denman	Stuart Ulmer	White Mountain A
Clare Casey - Appointment	11/11/2017 5:25 A	Clare Casey	Magdalena Bentle	Biltmore Surgical I
Corey Crumpton - Appointment	11/11/2017 5:25 A	Corey Crumpton	Anton Chew	Az Pain Centers of
Dianna Lancaster - Appointment	11/11/2017 5:25 A	Dianna Lancaster	Helena Wilcox	Biltmore Surgical I
Gabrielle Beck - Appointment	10/21/2018 10:36	Gabrielle Beck	Helena Wilcox	Az Pain Centers of
Kelsey Miranda - Appointment	11/11/2017 5:25 A	Kelsey Miranda	Dwayne Timmerm	White Mountain A
Monte Kong - Appointment	4/17/2018 7:56 PM	Monte Kong	Lily Connor	Spectra Eye Instit.

First Prev Page: 1 of 1 Next Last

OK Cancel

» *Figure 6 Select Test Data from the Microsoft CDS Data Store*

Once irVerify is loaded, the rules can be executed across the complete dataset, which may include additional data queried from within the rules themselves. All results are available for examination, including data changes, notifications, and the rule execution trace log. Using irVerify, rule authors can be confident that their business logic is accurate, optimized, and ready to run in a production environment.

The screenshot shows the InRule application interface. At the top, there is a ribbon with tabs for 'File' and 'Home'. The 'Home' tab contains several groups of icons: 'Data' (Load, Save), 'Test' (Apply Rules, Reset), 'Rule Tracing' (Enable, Open in New Window, Open in irAuthor, Save), 'General' (View Entity State, Performance Statistics), and 'Regression' (Create Test). Below the ribbon, a session tab 'Session 2 [Active] X' is visible. The main area displays a table with two columns: 'Name' and 'Value'. The table contains various patient attributes and their values, such as 'Birthday' (12/2/1976 12:00:00 AM), 'Age' (41), 'FullName' (Bryon Denman), 'MedicalRecordNumber' (MRN6112-4134), 'InsuranceType' (HMO - Bronze Plan), and 'AppointmentStatus' (Checked-In). Below the table, there is a 'Notifications [1]' section with a warning message: 'Warning: Patient insurance is not verified, please complete the verification task'. At the bottom of the notification section, there are buttons for 'Copy All', 'Copy Selected', and 'View All'.

Name	Value
Birthday	12/2/1976 12:00:00 AM
Age	41
FullName	Bryon Denman
FirstName	Bryon
LastName	Denman
MedicalRecordNumber	MRN6112-4134
Gender	Male
Encounters (1 member)	Add
ReferralRequests (no members)	Add
InsuranceType	HMO - Bronze Plan
CoverageStartDate	10/1/2017 12:00:00 AM
CoverageEndDate	9/30/2019 12:00:00 AM
CoverageVerified	<input type="checkbox"/>
CoverageStatus	Active
PractitionerType	
AppointmentStatus	Checked-In
Practitioner	Contact:2 New
AppointmentEMRStart	10/22/2018 5:36:24 AM
AppointmentCreationDate	7/16/2018 5:36:24 AM
PatientInstruction	Enter through location specified to office
Tasks (1 member)	Add

Notifications [1]

Warning: Patient insurance is not verified, please complete the verification task

Figure 7 irVerify Rule Test Scenario, Populated with Data

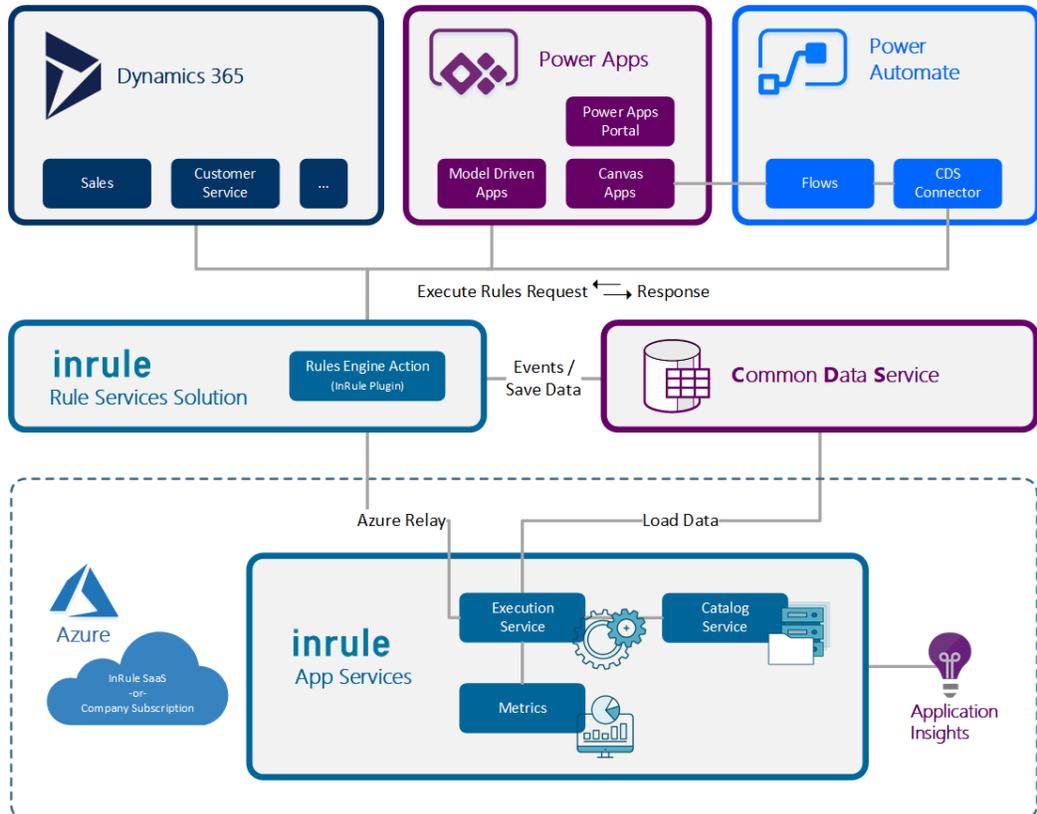
## Deploying Rules to an Environment

When the rule application has been developed and tested, the next step is to deploy it to the InRule Catalog hosted in Azure. This makes the rules available for execution directly from Dynamics 365, Power Apps, or Power Automate. The solution architecture that enables the integration of InRule with the Power Platform is described in the following sections.

## InRule and the Power Platform

With Microsoft unifying its low-code vision across Dynamics 365 and the Power Platform, the opportunities for InRule integration have never been greater. InRule’s event-driven and on-demand decision capabilities can now extend beyond Dynamics to drive processes within Power Apps and Power Automate. This is made possible by the CDS, which is the backbone data integration layer across Dynamics 365 and the Power Platform. InRule is designed to work directly with CDS enabling seamless data integration and bidirectional interoperability.

The following diagram depicts the integration possible between InRule, Dynamics 365, Power Apps, and Power Automate in the Azure cloud environment.



» Figure 8 Fully Integrated Business Rule and Low-code Application Platform

The components of the Power Platform that most directly benefit from integration with InRule are Power Apps and Power Automate. Within Power Apps, InRule can be leveraged across Model-Driven Apps, Canvas Apps, or the Power Apps Portal, but there are considerations for each that are explored in the following sections.

Rule execution requests may originate from any number of end business applications hosted on the Power Platform, which get handled by the InRule plugin and, in turn, passed to the Rule Execution Service via a secure Azure relay. The enriched rule execution results, including changed data and notifications, are returned to the plugin to be persisted and passed back to the caller.

This seamless round-trip integration is enabled by the event and data management services provided by CDS. Having common data services residing across all platforms simplifies deployment and establishes comprehensive integration options.

## InRule Integration Methods

There are several options when it comes to choosing how to integrate InRule with Dynamics 365 and the Power Platform. Beyond Dynamics, virtually any Power App or Power Automate Flow can benefit from decision automation. With InRule and the Power Platform, citizen developers have maximum flexibility to implement end-to-end business processes, such as claims processing, adjudication, eligibility assessment, scheduling, and countless other scenarios. Users can choose from the several InRule integration methods available to design a solution to suit their specific needs and rapidly respond to future change requests.

Method of Rule Execution	Dynamics 365	Power Platform			
		Power Apps			Power Automate
		Model-Driven Apps	Canvas Apps	Power Apps Portal	
CDS Events	●	●	●	●	●
Rules Engine Action (InRule Plugin)	●	●	◐	◐	●
Run Rules Button	●	●			
Workflow Activity	●	●			
Form Events	●	●			
JavaScript	●	●			

» Figure 9 Available InRule Integration Paths

**CDS Events.** The Rule Services Solution contains the InRule plugin that can act as a handler for events fired by the Common Data Service. By registering new steps on Create or Update events, rules will execute against the data associated to the event as it occurs. Using this model, rules can make additional data changes, contribute to transaction validation, and cancelation logic.

For example, during medical case management, a practice may wish to monitor prescriptions to prevent potential abuse. Perhaps the application should flag when the number of patients (or a percentage of patients) with specific prescriptions exceeds a set limit. When that trigger event occurs, the application would call the case review rules, within InRule, to analyze case records and other factors to determine an action.

**Rules Engine Action.** The InRule plugin contained within the Rule Services Solution has a preregistered step called the Rules Engine Action. This custom action allows rule execution to be called as a web endpoint via the CDS API. This web endpoint serves as the underlying transport for the various InRule integration methods. The Rules Engine Action centralizes communication with InRule in a standardized manner that can be called from other plugins, JavaScript, workflow activities, or the CDS Connector.

**Run Rules Button.** This is the easiest and most direct way to engage InRule functionality in an on-demand basis. The Rule Services Solution places the Run Rules button on the ribbon of each

Dynamics 365 or model-driven Power App. A click of the Run Rules button does exactly that: it executes rules against the current form data based on user configured settings.

Executing rules via the Run Rules button allows for the display of Informational, Warning, and Error notification types, which provide rich feedback to end users. Multiple rulesets can also be configured for interactive selection giving users the option of choosing which rules to run for a given record or dataset. When using the Run Rules button on entity view pages, rules will execute across all selected records in a serial manner. Returning to the prescription example, an auditor might want to look at provider practice groups across specific types of treatments and flag possible abuses by using the rules button.

**Workflow Activity.** Rules can be run directly from a workflow; the InRule workflow activity passes the appropriate parameters to the rule execution service and returns the same notifications and results as other InRule execution methods. Cancellation is also available from the workflow activity based on returned validation or error messages.

In medical case management, rules that define which specialists are included on a patient treatment team can be very complicated. Rather than customizing the workflow in Dynamics 365 or Power Platform using code, developers can pass the appropriate request parameters to InRule and incorporate the rules governing the treatment team into the workflow.

To run rules against multiple records, workflows can be leveraged from an entity view page against the selected records. This action is similar to using the Run Rules button on the entity view page. One notable difference is that workflows allow for the application of additional process logic before and after executing rules.

**Form Events.** These client-side events allow for the triggering of rule execution via common form events such as OnSave. The `invokeCustomAction.js` web resource provides a helper function called `executeRulesOnEvent()`, which can be registered to an event on the form.

When this JavaScript function is registered to the OnSave event, it triggers the rules configured for that event. If rules return any validation errors, the function automatically cancels the save operation and displays the corresponding messages in the notification pane.

The primary difference when choosing between the two supported event models is that Form Events are handled client-side in Javascript while CDS Events are handled server-side within the InRule plugin. While both approaches accomplish similar rules integration, each represents subtle differences that are fully detailed in the InRule technical documentation.

**JavaScript.** In certain scenarios requiring advanced integration patterns it can be relevant to invoke InRule from custom JavaScript. InRule exposes this capability with the `executeRules()` function from the `invokeCustomAction.js` web resource. When calling InRule from custom JavaScript, the integration behavior can be controlled by the managed rule configurations from the InRule solution or overridden with a JavaScript-based configuration object. Typically, this method applies to more advanced scenarios when client-side customization is required before and after invoking the rules engine.

## Additional Integration with Power Automate

Power Automate is the Microsoft low-code solution for building custom workflows and business logic. Power Automate flows interact with InRule through the CDS connector, which provides access to data and CDS-specific functions, such as calling custom actions.

Using the built-in CDS connector, Power Automate can call the Rules Engine Action (custom action) included in the InRule solution and pass all the information required to execute rules. This action kicks off the same plugin logic that's used by the Run Rules button and entity events within Dynamics 365 and returns the same set of information.

Once the rules perform all the required business process logic, the custom action can fire and move on, or use the rule output later in the flow. For example, rule notifications can be incorporated into an email, such as approving or denying a procedure, or construct downstream branching logic in the flow based on the rule output.

Because Power Automate integrates natively with Dynamics 365 and the Power Platform, it can call InRule from virtually any scenario, providing additional flexibility for processes and workflows.

Canvas Apps, when used in conjunction with Power Automate, provides a robust set complimentary functionality. Canvas Apps allow users to drag and drop elements onto a canvas, including a button that can invoke Power Automate which in turn can execute rules. Both Power Automate and InRule will have access to all of the relevant data available to the Canvas App by interfacing with CDS. With Power Automate and CDS, Canvas Apps can benefit from rule execution in a similar manner as Dynamics 365 and Model-Driven Apps.

## Conclusion

In this paper, we discussed how InRule helps businesses define and deploy rules using consistent, transparent logic without hiding the decisions in code. InRule's technology is ideally suited to enable organizations to leverage Microsoft's Power Platform while still being able to make automated, highly complex, personalized, and in-context decisions—in real time—based on the rules that run their business.

Working with Dynamics 365, the Power Platform, and InRule simplifies and speeds up the authoring and implementation of business rules that support transparency and accountability in customer-centric decisioning.

For more information, see:

- [Dynamics 365](#)
- [Power Platform](#)
- [Power Automate](#)
- [InRule](#)
- [irAuthor](#)
- [irX](#)
- [Deployment Guide](#)

InRule integration with Dynamics 365 and the Power Platform enables organizations to focus on their customer and develop applications and workflows that transform data into meaningful interactions.

Better business apps begin with InRule, Dynamics 365 and the Power Platform. To get started, [find InRule in the Microsoft AppSource today](#).

InRule, irAuthor, irVerify, and irX are registered trademarks of InRule Technology, Inc. All other trademarks and trade names mentioned herein may be the trademarks of their respective owners and are hereby acknowledged.