Brought to you by ServiceNow

servicenow

Low-Code Apps

for dummies®
A Wiley Brand

ServiceNow Special Edition

Digitize your workflows fast

Unlock productivity across your enterprise

Create amazing experiences

Chuck Tomasi

Brad Tilton

About ServiceNow

ServiceNow makes work, work better for people. Its cloud-based platform and products streamline and simplify how work gets done. ServiceNow delivers digital experiences that help people do their best work fast, creating great employee and customer experiences. ServiceNow (NYSE: NOW) works for you. To learn more, visit **www.servicenow.com**.

Low-Code Apps

ServiceNow Special Edition

by Chuck Tomasi and Brad Tilton

for dummies®
A Wiley Brand

Low-Code Apps For Dummies®, ServiceNow Special Edition

Published by
John Wiley & Sons, Inc.
111 River St.
Hoboken, NJ 07030-5774
www.wiley.com

ISBN: 978-1-119-82038-3 (pbk); ISBN: 978-1-119-82039-0 (ebk).

Some blank pages in the print version may not be included in the ePDF version.

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

Publisher's Acknowledgments

Some of the people who helped bring this book to market include the following:

Project Manager and Development Editor: Carrie Burchfield-Leighton

Sr. Managing Editor: Rev Mengle

Acquisitions Editor: Ashley Coffey

Business Development Representative: Cynthia Tweed

Special Help: Donna Tomasi, Todd Zambrovitz

Table of Contents

Introduction

When it comes to digital transformation, many organizations need to move faster to meet changing business requirements — and it takes a lot of software in the form of applications (apps) to accelerate and improve how work gets done. In the past, IT was the go-to group for getting those apps built and delivered. These days, IT teams are pretty maxed out, and project backlogs are commonplace. The good news is that you can increase your app delivery capacity by empowering more people to build applications with less complexity. And that's exactly the premise behind low-code development.

About This Book

This book explains how anyone can automate, extend, and build digital workflow apps across their organizations by using the low-code capabilities of ServiceNow Creator Workflows. Powered by the Now Platform, ServiceNow Creator Workflows combines the capabilities of App Engine and IntegrationHub so your organization can tap into the benefits of low-code application delivery.

Low-Code Apps For Dummies, ServiceNow Special Edition, consists of six chapters that explore the following:

- Creating a plan for your app (Chapter 1)
- Basic data setup techniques (Chapter 2)
- Creating an amazing experience for your app's users (Chapter 3)
- Building logic to unlock productivity (Chapter 4)
- Additional low-code capabilities to add to your app (Chapter 5)
- Low-code tips and tricks from the experts (Chapter 6)

Foolish Assumptions

We made some assumptions about you, our reader, when we wrote this book. Mainly, we assume the following:

- **You're a subject matter expert in your role.** You may be a business analyst, specialist, or senior member of a team. Whether you're in the accounting, legal, marketing, or safety department, you know your stuff.
- **You're in an organization that's changing.** Business requirements evolve. Either you're fortunate enough to have to scale quickly, or you're being asked "to do more with less." Either way, yesterday's techniques and technologies just aren't effective today.
- **You don't create applications for a living.** You didn't go to school for a computer science degree, and you've likely never written any code.
- **You recognize that you have ad hoc processes.** These processes may use email, spreadsheets, or perhaps even paper (gasp!). You also recognize that these processes could be improved with digital transformation.

Icons Used in This Book

Throughout this book, we use icons in the margins to draw your attention to certain kinds of information. Here's what the icons mean:


REMEMBER

This book is a reference, which means you don't have to memorize it, and there won't be a test on Friday. But when we tell you something so important that you should commit it to memory, we use the Remember icon.


TIP

Whenever you see the Tip icon, you can be sure to find some useful nuggets of information that save you time or money or just make your life a little easier — at least when it comes to developing apps.

WARNING

The Warning icon alerts you to things that could cause you big headaches. Think of these as orange cones in the road, warning you about an open manhole cover. Sure, you could ignore them, but you may take a nasty fall.

Beyond the Book

This book focuses on the conceptual steps to building an app and points out many low-code capabilities within ServiceNow Creator Workflows that enable you to build those apps, but we don't have room for detailed "how-to" information. If you want even more information, check out the following resources:

- **servicenow.com/workflows/creator-workflows.html**: Visit the ServiceNow Creator Workflows page to discover more about building connected digital workflow apps with a low-code platform.
- **developer.servicenow.com/builder**: The Service Now Builder Page has plenty of beneficial links to help you get started fast. Get a free personal developer instance, helpful videos, online learning plans, and more.
- **devlink.sn/builder-videos**: This video series takes you through the "how-to" steps of building an example app by using the concepts and capabilities presented in this book.
- **knowledge.servicenow.com**: This link takes you to ServiceNow's annual Knowledge Conference page. The CreatorCon event at Knowledge contains many hands-on labs tailored to you, the builder. Sign up to stay informed about Knowledge keynotes, speakers, and events.
- **docs.servicenow.com**: The ServiceNow product documentation site has full documentation to the various platform capabilities mentioned in this book.
- **community.servicenow.com**: If you find yourself in need of help, the thousands of subject matter experts in the ServiceNow community are eager to offer help on a variety of topics.

ServiceNow invites you to have a discussion with your ServiceNow account team for other opportunities like hands-on workshops, webinars, and more. If you don't have an account representative yet, no problem. To connect with a rep, visit www.servicenow.com/contact-us.html.

IN THIS CHAPTER

- Asking the right questions before building an app
- Knowing what goes into good app design
- Identifying what you need to get started building an app

Chapter 1
Beginning with a Plan

You probably wouldn't head out on a road trip without at least a general idea of how to get where you're going — at least not if you want to get there anytime soon. Planning is essential in life, as well as in app development.

In this chapter, you discover the importance of planning and the key questions to ask yourself before you start building your app. You also find topics to consider to ensure you achieve the best possible outcome for your app.

Before You Build: Asking the Right Questions

To help you determine how to best utilize the features in Creator Workflows to build an app that maximizes business value for your organization, take a look at the following questions and consider your answers:

- **What are the goals, objectives, and outputs of your app? In other words, what business problem are you trying to solve?** Without a specific business objective, you'll have difficulty measuring the success of your app or justifying its continued use within the organization.

TIP

Before you start building, begin with the end in mind. Understanding and visualizing (virtually or on a whiteboard) your desired solution helps determine the remaining steps in building your app. Often, the outputs are the drivers for the inputs. If you're trying to speed up a process, for example, knowing your output metrics can help make clear what to measure. If you're managing assets, perhaps cost and location are more important than the minute details of each item. Identifying your goals and objectives ensures you can manage conversations with key stakeholders so your app is specifically addressing your desired business outcomes.

Here's an example of a clear objective: *Reduce the time it takes to route and approve time-off requests from five days to less than one day.*

- **Are you taking a spreadsheet and turning it into an app in ServiceNow, or does the app exist somewhere else?** This question impacts your approach for building the app because there are different tools within the platform to support your efforts.

TIP

Take this opportunity to review and revise your process. Too many times, processes are dictated by limitations of legacy tools. Don't cripple your new app by trying to make it work like the old one did. After all, if the old app worked perfectly, you wouldn't be building a new one, right?

- **Who will be using your app?** Identifying your target audience has a direct impact on the features your app will provide, the data it will capture, and the interface you'll need to provide for your app.
- **Do you want everyone to have the same ability to see and edit fields, or will some people need more or less access than others?** Security is a significant and ever-growing concern in most organizations, so identifying who has access to what during the planning stage is a critical step in app development.
- **What will the users do with the app?** Will they be providing information, collecting information, routing information, requesting information, looking up information, and/or collaborating on information? Identifying these actions establishes the features and functions you need to build into your app.

- **Where is the data coming from?** One of the most common assumptions is that data will be entered by people. Some data (like users, departments, and locations) may already be available within your ServiceNow instance (your very own installation of ServiceNow software in the cloud). You may also find that you require data from an external data source that you need to import.

  When necessary, leverage existing data sources to avoid data entry duplication and make sure your app has the data it needs to meet its business objectives.


TIP

- **How will people interact with your app? Will they use desktop computers, mobile devices, or both?** Understanding how people access your app impacts how your app will function. Will they take action with a swipe of a finger or click of a mouse?
- **Can you walk through one or more example use cases or scenarios?** Walking through an example use case, or "day in the life of," is a great way to discover an app's requirements.
- **Is there an existing app or template in ServiceNow that already does (most of) what you need?** Why reinvent the wheel? If there is an app or app template that does what you need, or close to it, look at the possibility of using or extending that existing app.

  Many organizations think that their processes are unique when they're actually pretty similar to what other organizations have done before. Take advantage of that similarity.


REMEMBER

- **How will you measure the success of your app?** If your app is meeting a business purpose, you may want to provide reports showing usage, adoption, and key performance indicators (KPIs) associated with the app to show outcomes achieved.
- **Is this a good fit?** Not every app idea makes for a good fit for a Creator Workflow. In general, your app is a good fit if it involves
  - Simple forms
  - Task management
  - Repeatable processes
  - Excel-driven processes
  - Request fulfillment

- Third-party integration
- Orchestrating multiple systems

If your app involves the following, then ServiceNow probably isn't what you need:

- Unstructured data
- Graphics processing or streaming video or audio
- Unrepeatable processes


TIP

Make sure your app is a good fit for ServiceNow before you start building.

Making Permanent Decisions

When you're building an app, you'll inevitably take some steps that are irreversible. You need to be aware of what these irreversible steps are so you can plan in advance and make the right moves.

Deciding where to build your app

Proof of concept (PoC) app builds can be built in a personal developer instance that you get from the developer portal (`developer.servicenow.com/builder`). These instances are named something like `dev12345.service-now.com`.


WARNING

You can rebuild PoC apps, but don't import them from your personal developer instance into your organization's instance. There is information included with your app that indicates where it was built. If you bring the app over from your personal developer instance, life will be a lot harder when you try to get your app in to production.


TIP

Apps that your organization will actually use (for example, production apps) should be created in your organization's developer instance so the app can follow your organization's testing and deployment process. See your ServiceNow System Administrator for more details about which instance to use for an app that will eventually be deployed to your organization's production instance.

Naming your tables and fields

After your app is created, you'll most likely be creating new tables and fields for it. Tables and fields have both labels (displayed in your browser and mobile user interface, or UI) as well as internal database names. Labels can be edited and even translated later, but internal database names can only be edited at creation time.


TIP

For tables, a label like "Safety issues" may produce a name of `x_snc_safety_safety_issues`. For consistency, use singular table names. ServiceNow automatically produces plural labels where needed. Also, avoid redundancy in the table name; `x_snc_safety_issue` proves a lot less troublesome than `x_snc_safety_safety_issue` when you're maintaining your app down the road. Similarly, you may be tempted to build fields with verbose labels, such as "How many widgets do you require?" This translates into a field named `how_many_widgets_do_you_require_` (because spaces and symbols become underscores in the database). This field label is troublesome for users because it may not be displayed as expected and developers will have to deal with an awful field name in their scripts. Instead, consider just labeling the field "Widgets" to create a field called widgets.


REMEMBER

You can always relabel, but you can't rename. If you want to provide a longer description, ServiceNow offers hover-over tips and clickable links.

Identifying the Prerequisites for Building an App

Before building your app, you need the following:

- **A ServiceNow instance:** You can get one for free at the ServiceNow developer portal (`developer.servicenow.com/builder`).
- **An admin or the `sn_app_eng_studio.user` role in that ServiceNow instance:** The latter is a role with fewer privileges than the admin role, but it still allows for app development.

TIP

The ServiceNow developer portal has something to offer all skill levels when it comes to solving real business problems using Creator Workflows, so don't let the name *developer* throw you off. Take a look at the low-code builder content. There you can find the following *free* perks:

- **A personal developer instance (PDI):** You can use your own instance running the supported ServiceNow release of your choice. Use your admin-level access to configure the instance and make amazing apps.
- **Early access:** Developer program members get access to the latest ServiceNow releases before they're generally available to the public.
- **Training:** Gain access to free learning plans, best practices, videos, and training modules.
- **Online and in-person events:** As part of the developer program, you're invited to ServiceNow developer events such as CreatorCon at the company's annual Knowledge Conference, as well as virtual and local events like hackathons, hands-on workshops, labs, meetups, and much more.
- **Community:** Get access to developer-oriented forums designed to help you build better apps. You can connect with and get guidance from other ServiceNow developers through online forums and in-person meetups.

IN THIS CHAPTER

- Seeing what tools you have at your disposal
- Working with tables
- Making the most of fields
- Paying attention to other table creation details

Chapter 2
Storing Your Information

After you've planned your app (see Chapter 1), you're ready to build your tables to store your data. In addition, you'll create fields in tables, possibly loading the table(s) with data and making sure the right people can access that data.

Getting to Know Your Toolbox

When it comes to storing your information, you have a few tools at your disposal:

- **App Engine Studio (AES):** AES provides you with a guided experience to create everything you need for your low-code app. You can begin with a template or start from scratch. Building the tables; importing spreadsheets, workflows, and user experiences; and managing security are fast and easy with AES.
- **Studio:** If you want to dig in deeper to some of the additional capabilities, Studio keeps track of your app's components (or files). Among developers, this is known as an integrated development environment (IDE).

- **Now Experience UI Builder:** UI Builder allows you to create workspace and portal experiences using a drag-and-drop interface. From simple page layout to advanced component configuration, UI Builder offers a lot.
- **Flow Designer:** Flow Designer enables process owners to use natural language to automate approvals, tasks, notifications, and record operations without having to code.

TIP

Be sure to work with your account team when considering building apps. Some capabilities may require additional licensing.

Making Choices about Your Tables

For creating tables, ServiceNow offers three methods:

- **Upload a spreadsheet:** Use the spreadsheet columns to define your new fields and, if you want, import the data.
- **Create from an existing table:** Also known as *extending a table,* you can leverage an existing table to instantly create fields, logic, and more. This is a great way to accelerate your app creation.
- **Create a table from scratch:** You can build a new table and fields from the ground up. This method gives you total control over what information you want to store, but it may require a bit more work than the other two methods.

The ServiceNow app screen with your choices of methods is illustrated in Figure 2-1.

REMEMBER

Uploading a spreadsheet is available only through AES or when creating a new application in Studio.

WARNING

When you create a table from scratch, you can't go back and make it an extended table. Likewise, when you create an extended table, you can't "unextend" it later. If you made a mistake when you started out and you want to change it now, you need to create a new table and migrate your data.

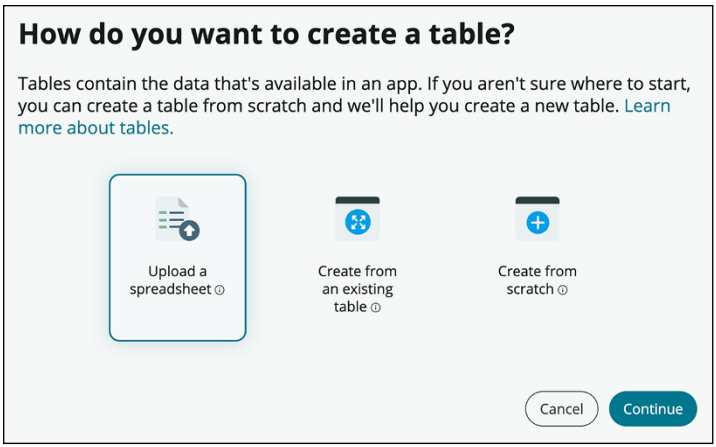


FIGURE 2-1: Creating a table in the ServiceNow app.

If you have doubts about whether to create from scratch or extend, it's generally better to extend a table and not need the available fields and functionality than it is to realize down the road that you need them and you don't have them. Consider your options carefully before creating the tables and fields that make up your data model, and remember, not every table should be extended from another table. We cover each of these table creation methods in more detail in this section.

Extending a table

When you extend a table, your new table inherits all the fields and functionality from the table you're extending, saving you time. By far the most common table to extend in ServiceNow is the task table.

TIP

To determine if you want to extend a table, use the decision tree in Figure 2-2.

An additional, yet important benefit of extending a table is roll-up reports. ServiceNow provides several tables already extended from the task table. Viewing data that all shares the same base (task) table is a no-brainer. A common example is when an employee wants to see all the work assigned to her. She only needs to look at the task table and filter on the Assigned To field to see tasks across multiple processes. Now you come along with a killer expense report app and choose to extend the task table. Automatically the expense reports assigned to an employee are added

to the list of tasks without any additional work on your part. If you didn't choose to extend the task table, employees would have to look at multiple lists from different tables to see all the work assigned to them.

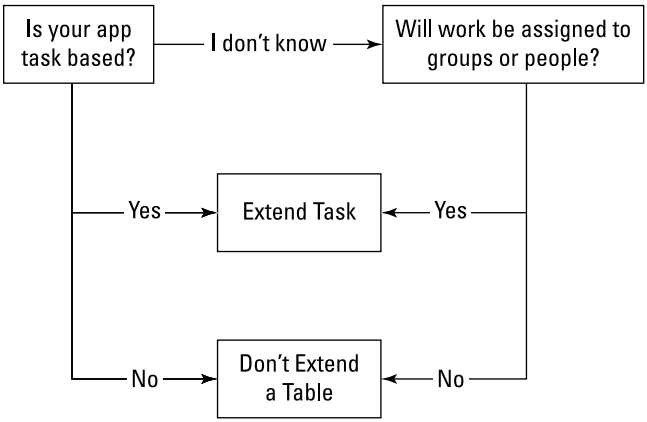


FIGURE 2-2: Extending a table can accelerate your app build process.

If you determine from Figure 2-2 that extending an existing table is a good option for you, simply identify which table to extend and proceed to the next screen. From there, you can get familiar with which fields you inherited and add any fields you need to your new table.

TIP

When you extend a table, you have a number of fields to choose from (instead of creating new fields). Before creating a new field, check to see if there's an existing field that may meet your purposes simply by changing the field's label. Note that the purpose of the field should be similar to the purpose of the field in the base table.

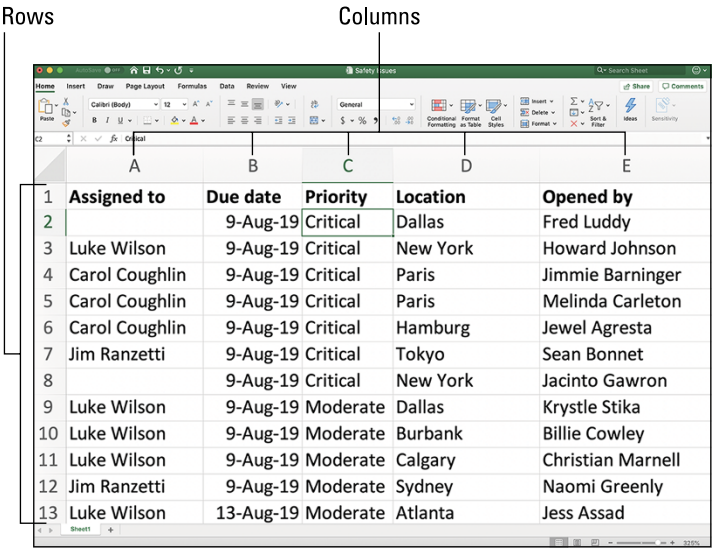
You may want to extend a table in the following circumstances:

- **You have work that needs to be assigned to someone.** This would be a good time to look at extending the task table because it already includes fields to assign to a group and user.
- **You have an asset that has similar, yet specific properties to something you already own.** Let's say you want to track tablets. They share many of the same fields as

computers, but they have some unique aspects. Extending the computer table would get you most of the fields you need to track tablets.

Uploading a spreadsheet

If you're creating an app based on a spreadsheet, each worksheet is likely to map to a table in ServiceNow, each column may become a field in that table, and each row may become a record in that table (see Figure 2-3 and 2-4).



	Assigned to	Due date	Priority	Location	Opened by
2		9-Aug-19	Critical	Dallas	Fred Luddy
3	Luke Wilson	9-Aug-19	Critical	New York	Howard Johnson
4	Carol Coughlin	9-Aug-19	Critical	Paris	Jimmie Barninger
5	Carol Coughlin	9-Aug-19	Critical	Paris	Melinda Carleton
6	Carol Coughlin	9-Aug-19	Critical	Hamburg	Jewel Agresta
7	Jim Ranzetti	9-Aug-19	Critical	Tokyo	Sean Bonnet
8		9-Aug-19	Critical	New York	Jacinto Gawron
9	Luke Wilson	9-Aug-19	Moderate	Dallas	Krystle Stika
10	Luke Wilson	9-Aug-19	Moderate	Burbank	Billie Cowley
11	Luke Wilson	9-Aug-19	Moderate	Calgary	Christian Marnell
12	Jim Ranzetti	9-Aug-19	Moderate	Sydney	Naomi Greenly
13	Luke Wilson	13-Aug-19	Moderate	Atlanta	Jess Assad

FIGURE 2-3: Spreadsheets contain rows and columns to store data.

To import your spreadsheet, AES offers a step-by-step approach:

1. **When you create tables in AES, simply click Upload a spreadsheet (refer to Figure 2-1).**
2. **Drag and drop your spreadsheet.**
3. **Define the field types needed (see the "Creating Fields" section later in this chapter).**
4. **Optionally import the spreadsheet data.**

It's that simple.

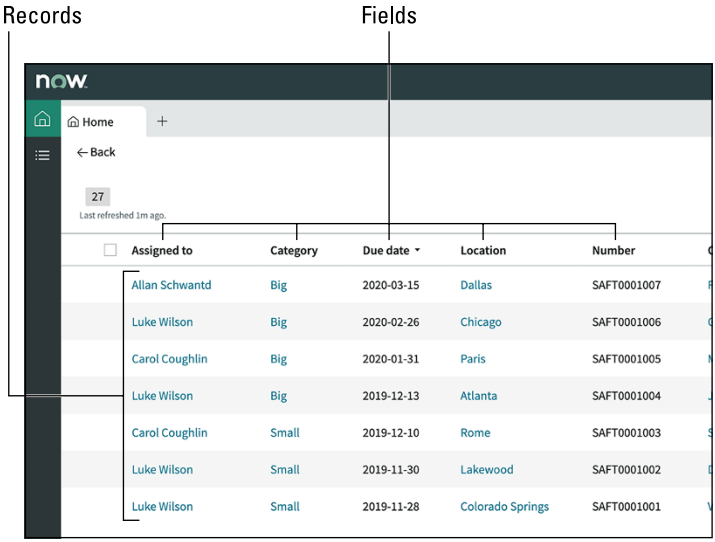


FIGURE 2-4: ServiceNow tables use records and fields to store data.

Creating a table from scratch

Another option for creating a table is to create each field yourself. If you determine from Figure 2-2 that your best option isn't to extend a table, then click Create from Scratch (refer to Figure 2-1).

The screen presents an interface that allows you to choose your field labels, types, and other properties, similar to the other two options to create the fields you need in your table. It's a lot like extending a table (see the earlier section in this chapter titled "Extending a table"), but you don't get any existing fields beyond the standard system fields (we cover this in more detail in the later section "Field attributes").

Creating Fields

After you've created a table, you need to add fields to it. ServiceNow has many different field types with built-in validation. Choose the one that best fits that field's data type.

WARNING

You can easily make plain-text (string) fields where people can enter anything, but doing so can result in bad and inconsistent data that's difficult to use. For example, if you have a field on your table for someone's name, you use a plain-text (string) field and end up with data like you see in Figure 2-5.

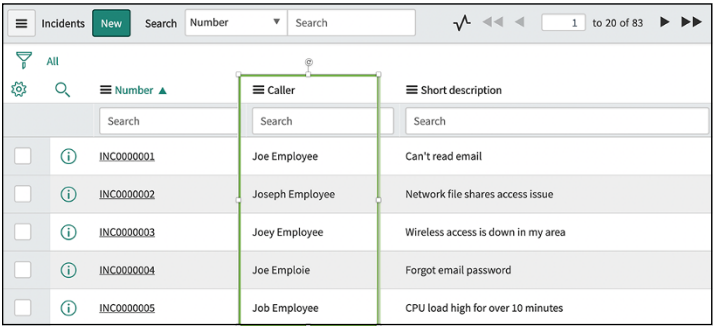


FIGURE 2-5: Using the wrong field type can lead to data inconsistency issues.

But if you use a reference field instead of a plain-text field, you get data that looks like Figure 2-6. Much better.

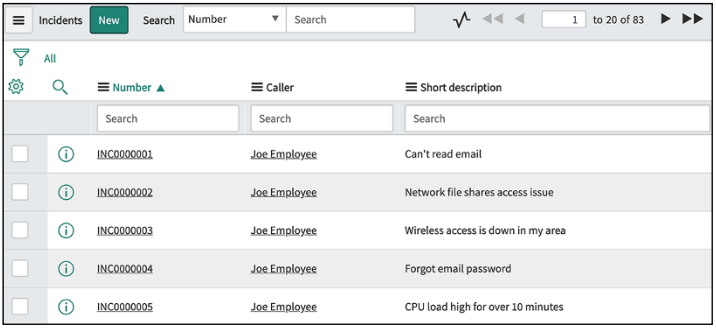


FIGURE 2-6: Reference fields are one way to standardize data.

REMEMBER

You can use reference fields to make your data consistent (or normalize it) by referencing an existing table in ServiceNow. ServiceNow has more than 2,000 tables at your disposal. Table 2-1 lists some commonly used tables for building an app in ServiceNow.

TABLE 2-1 Commonly Used Tables

Label	Name	Description
User	sys_user	List of all ServiceNow instance users.
Location	cmn_location	List of all user locations. Users are typically associated with a location.
Group	sys_user_group	List of all the groups. Users are typically associated with groups and inherit any security roles associated with those groups.
Company	core_company	List of companies that interact with your organization.
Role	sys_user_role	List of security roles in the instance. Some will be default roles; others will be created by your organization.
Task	task	The common base table that gets extended. It has fields and functionality related to assigning work across teams and individuals, managing the state or the task, and other functions.

While a reference field can normalize your data, other fields can be used for specific types of data. The complete list of field types can be found on the ServiceNow Product Documentation site at `devlink.sn/field-types-docs`, but Table 2-2 lists some common field types.

TABLE 2-2 Commonly Used Field Types

Field Type	Notes
Integer	A freeform input field that accepts number values only. Use this field type if the value will always be a number and you may be using it in calculations.
Currency	A freeform input field with a currency type. Use this field type when dealing with money.

Field Type	Notes
Phone number	A combination of drop-down list to select country format and freeform input for the number. Use this field type when you need to validate phone numbers.
Reference	A record picker. Use this field type when you want to reference a record from another table.
Choice	A drop-down list. Use this field type when you need a short list of options to present to the user.
Date	A date picker. Use this field type if you don't need a specific time.
Date/time	A date/time picker. Use this field type if you're comparing specific times or the exact time is important.
String	A freeform text field. Use this field type if no other field type fits your purposes.

Choice fields versus reference fields

A common question among new builders is, "When should I use a choice field and when should I use a reference field if both are great for normalizing data?" Here are two questions to ask yourself:

- **How many options are you offering?** Use a choice field if your list of options is fairly short (say, less than 10 to 15 items). For example, you may want your user to pick a color. The list you propose contains values Red, Green, Blue, Yellow, Orange, and Silver. That's perfect for a choice field.

  If your list has more than 15 items, you're probably better off with a reference field, so you don't cause the user to have to scroll forever. Reference fields make the user experience better in this case by offering a type-ahead feature. For example, in a list of names, if the user starts typing "br," she may be presented with options for Brad Tilton, Brett Oliver, and Brian Murray; then she can choose the correct value. The user can also use the magnifying glass icon to bring up a list and, optionally, filter to choose a name from that list.

- **What are your data values?** In the preceding bullet about colors, the option for Red may actually contain a value of #FF0000. This relationship is fairly clear.

  Unlike choice fields (which offer one option to one value), reference fields can have additional information related to the displayed choice. The user would still pick Red from a list of car colors, for example, but the related record would have much richer information. For example, a value of #FF0000, a default distributor, pricing information, and more.

Field attributes

Each field can have various attributes. Some attributes are based on the field type, and others are common to all fields. Be sure to review the field types to determine if you want the field to be read-only, to be mandatory, to contain a default value, and more. How you set your field attributes can make a big impact on how users interact with your app.

Six fields, shown in Table 2-3, are automatically created for every table in ServiceNow. They contain auto-populated information about the table, like when it was created, when it was last updated and by whom, as well as a unique identifier for the table. These fields can't be manipulated.

TABLE 2-3 Default Fields in ServiceNow

Field Name	Database Name	Description
Created by	sys_created_by	The user who created the record.
Created	sys_created_on	The date/time that the record was created.
Updated by	sys_updated_by	The user who last updated the record.
Updated	sys_updated_on	The date/time the last record was updated.
Sys ID	sys_id	The unique identifier for the record. This is auto-assigned and unique throughout the instance.
Updates	sys_mod_count	The number of times this record has been updated since the record was created.

Putting the Finishing Touches on Your Tables

Creating a table is much like picking out a car — you may have your mind set on a make, model, and perhaps a color, but have you considered the accessories like the tires, the stereo system, the engine size, the interior style, the electronic gadgets? If you choose wrong, some of these can be swapped out or upgraded later, but others are permanent, and you're going to have to live with them.

With tables, it's much the same. This section explains some of those check boxes, drop-downs, and other fields you encounter when you create your tables.

Choosing a table label

The table label is used wherever someone interacts with a list or record for your table. Some options include the left navigation menu, at the top of a list or record, or in a pick list of tables. The table label is modifiable (and translatable) after your app is created.

Picking a table name

The table name is the database name. It's typically not displayed to your end-users, but it's the way in which ServiceNow interacts with the database. The table name is also prefixed with the app scope name.

WARNING

Consider table names carefully. After the table name is created, you can't change it, so we encourage you to review your table names before saving so they make sense later. If you have an app called Loaner to manage your loaner items and you create a table called Loaner Request, the default table name may be something like `x_snc_loaner_loaner_request`. Before you save your table, consider modifying the table name to `x_snc_loaner_request`. You, or your developers, will appreciate this if you need to add scripting to your app later.

Making your table extensible

You can make any table you create capable of being extended to other tables simply by checking the Extensible check box.

Say you're building an app to track vehicles. These vehicles could be anything from cars to trucks to electric bicycles. The vehicles have a certain number of common attributes like owner, number of wheels, date of purchase, color, and so on. These fields could be put in a base table that you later use to extend to other tables with specific attributes (or fields) of their own. For example, number of doors would be applicable to cars and trucks, but not bicycles or motorcycles so that wouldn't be a good candidate to put on your base table.

Auto-numbering your records

Auto-numbering allows you to add a sequential number to your record with a prefix. This prefix acts as a unique, human-readable designator so you can quickly find the record later. By checking the Auto-Number box when you create your table, you tell the system to create a field called Number with an associated character prefix and counter that gets automatically incremented with each new record.

Consider an app that tracks safety issues. It's much easier for someone to call in and refer to SAFT0010022 than it is to try searching for "that thing that happened in the break room where the wire wasn't plugged in to the doohickey."

REMEMBER

Not all tables need auto-numbering. The most common use of auto-numbering is for task-based records. Records that support a process (such as lists of people, locations, groups, or devices) are typically not auto-numbered.

Managing Access

In Chapter 1, we cover planning considerations and making sure everyone has the correct ability to see and edit your data. So as you create your tables, this is a good time to remind you to consider "who needs what."

There are four types of access: create, read, write, and delete. Each role you create may have different access. Consider granting the appropriate access based on the personas, or roles, your app requires. For example, you may want a user with an approver role to your app to have read and write access, but not create and delete access, whereas a user with an employee role may need to create as well as read and write.

When building your app with AES, there's a section labeled "Security" that makes it easy to create roles based on your personas, and then you can modify the access to create, read, write, and delete records based on those roles. You can also create more detailed security rules using Studio to define read/write controls on specific fields.

WARNING

Use delete access with caution. Deleting records isn't normally something you want to do because it can leave gaps in your data. Consider creating a true/false field named "Active" to simply deactivate records you don't wish to see using a simple filter.

IN THIS CHAPTER

- Getting to know the classic user interface of forms and lists
- Giving your users a mobile experience
- Providing a workspace
- Creating custom portals for your end-users
- Working with reports and dashboards

Chapter 3
Creating Amazing Experiences

How will people interact with your application? Creator Workflows offers several ways to allow your users to interact with your app. There are standard forms and lists, a native mobile application, a custom portal, and a more recent user interface (UI) that ServiceNow simply calls *workspace.* Each has its own merits and is well suited for a specific type of persona or the work he or she typically does.

When building your app, you should think about several design considerations. Will your app be accessed via desktop, mobile, or both? Is your target audience already using an application on ServiceNow where it's comfortable using forms and lists, or will it need a self-service type of interface? Will your audience be doing quick updates on the go, or does it work in ServiceNow the majority of the workday? This chapter has you covered.

Using Forms and Lists

The most common method of accessing data in ServiceNow is through forms and lists. A *form* displays information from one record in a data table, and a *list* displays a set of records from

a table. Users may interact with a form or list across multiple interfaces, such as workspace, portal, or the legacy view, but the following guidelines apply no matter the interface:

- **Keep the number of fields on a form to a minimum.** The more fields you have on a form, the longer it will take to load. Generally, users don't want to work with a long form either. You can use form views to create different sets of fields for different situations.
- **Use form sections to logically group fields together and keep users from having to scroll.** The top section of the form should contain the fields that are always needed or used, while the other form sections contain less frequently used fields.
- **Make sure fields appear in a logical order.** For example, a start date field should always come right before an end date field.
- **Use seven or fewer columns in a default list.** As tempting as it is to put a lot of fields on a list, users will have to scroll horizontally to see the "missing" columns, and that's just not a good experience.
- **Avoid using a reference field as the first item in the list view because it's shown as hyperlinked text.** Clicking the reference field redirects the user to the referenced record instead of the list record and results in a poor user experience. For example, when viewing a list of case records, the expectation (based on the way the majority of lists are configured) is to click the value in the first column. If your list is different, in that it takes the user to the case owner instead of the case details, that's off-putting for the user.

A poorly designed form has the following characteristics:

- There are no sections — it's one long form.
- Fields that don't require much space, like the date fields at the bottom, are taking up the full width of the form.
- The left and right sides of the upper form feel unbalanced with more items on the right than the left.
- Similar fields aren't grouped together (for example, "assignment group" and "assigned to" are nowhere near each other).

Nobody wants to use a poorly designed form, such as the one illustrated in Figure 3-1.

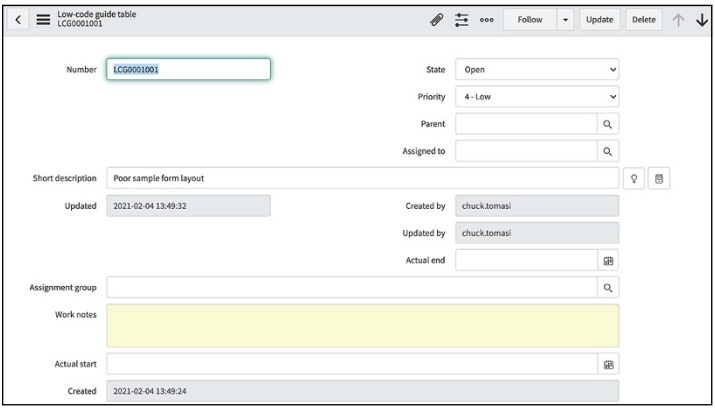


FIGURE 3-1: A poorly designed form.

By contrast, good form design, shown in Figure 3-2, can unlock productivity.

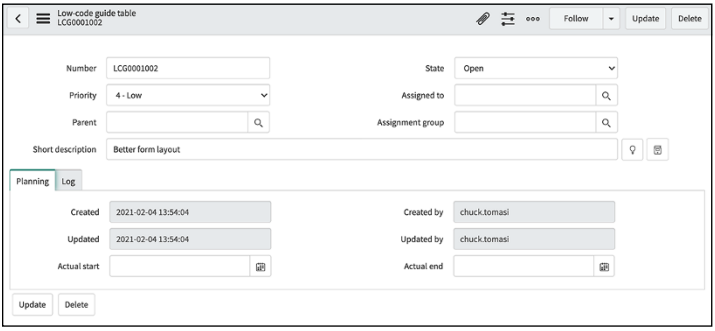


FIGURE 3-2: A well-designed form.

This form is well designed because it has the following characteristics:

- Fields are grouped together logically, like "assignment group" and "assigned to."
- Fields that don't take much space (date, choice, and reference fields) are placed side by side.

- Short fields are balanced left and right where possible.
- The form has been broken into sections for easier viewing and data entry.

Taking It Mobile

ServiceNow offers native mobile capabilities for Android and iOS users. If users require functionality like geolocation or offline access to your data, you can use one of the two native ServiceNow apps: Mobile Agent, for the agents and fulfillers, and Now Mobile for your end-users (see Figure 3-3). The development is done by creating a mobile application in Studio inside of the application you've been building. You don't need to learn iOS or Android development tools. The Now Platform takes care of the hard part, allowing you to focus on the logic and presentation of your app.

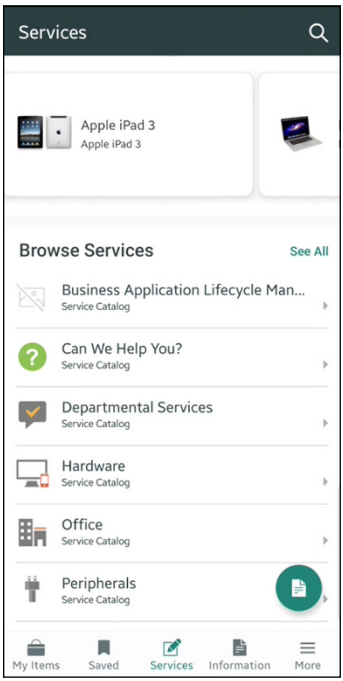


FIGURE 3-3: Access your data on the go with a mobile app.

WHAT MAKES A GOOD MOBILE EXPERIENCE?

Mobile apps aren't designed to be a mobile version of all desktop functionality. The best mobile experiences come from quick interactions. When creating a mobile experience, keep the actions simple to allow users to create and update records. Think about the mobile apps you use the most to hail a ride or shop online. You open the app, you make your request, and you're done in a few minutes. The idea of mobile is to make it quick and easy. Some people have referred to this as a *targeted micro-experience.*

Individual applets can be secured by role, as well as made available in offline mode.

If you use App Engine Studio (AES) to get your app started, clicking the mobile option allows you to easily create a mobile experience simply by choosing tables. You can also create mobile apps manually using Studio to allow users to interact with your data.

Working with Workspaces

You may find yourself with users of your app who practically live in ServiceNow to do their daily jobs. We refer to these users as *agents*. Agents may be people who fulfill requests, respond to cases, or address inquiries — theirs is a life of constant data flow. To make their jobs easier, ServiceNow offers Workspace.

Workspace is a suite of tools that provides agents, case managers, help desk professionals, and managers with an integrated and graphically intuitive user experience. Workspace features, shown in Figure 3-4, include

- A multi-tab interface to manage multiple cases or incidents
- Real-time handling of calls and chats via the Interaction Management System
- Task resolution assistance via Agent Assist

- Intuitive search capabilities to quickly find relevant content
- Heads-up display of contextual information to quickly get oriented to new tasks

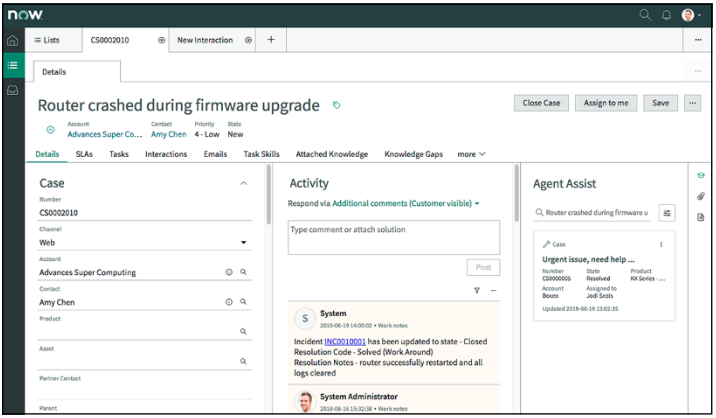


FIGURE 3-4: Easily manage multiple interactions with Workspace.

By default, Workspace is active for all instances.

Building a Portal Experience

If during the planning phase you decided that your application has a Requestor or Self-Service user, you may want to create an easy-to-use portal for them to find and access the information they seek. You can use the Now Experience UI Builder (we introduce this in Chapter 2) to quickly create and edit pages for your portal experience. Creating pages for your portal experience is the same as creating pages for workspaces.

TIP

To jumpstart your page building, use the start from a page template option when creating a new page in UI Builder.

Using Reports and Dashboards

REMEMBER

Most applications will have some level of reporting requirements. Reports should be created with actions in mind and be built to drive change. The in-platform reporting tool in ServiceNow is powerful and easy to use. You can start by simply clicking

a list column heading to make a bar or pie chart, or use the wizard interface to guide you through more complex options (see Figure 3-5). With great power comes great responsibility.

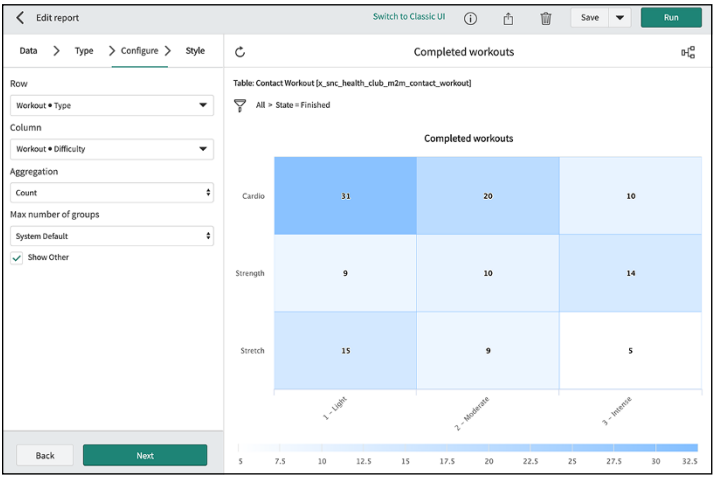


FIGURE 3-5: The built-in report builder offers a wizard experience for easy report generation.

Here are some guidelines to follow when creating reports:

- Be careful when reporting on large tables — it could have a performance impact on your ServiceNow instance. Make sure you're filtering by date range or another limiting criteria rather than showing all records in the table.
- When grouping records in a report, try to avoid grouping by fields that contain many possible values — it could impact performance.
- If running your report gives you a Long Running Transaction Timer message and takes a long time to run, consider adding more data filters to reduce the report run time.
- If someone needs a report daily or weekly, consider scheduling it to be sent via email.

The Now Platform reporting capabilities offer a wide variety of report types from simple bar graphs to heat maps to geographical maps. When you view a report in ServiceNow, the data is live — you can click a column in a chart and instantly view the underlying

records that make up that data. This is far more advantageous than exporting data to a third-party application.

You can also use dashboards to show multiple reports on one page like the example you see in Figure 3-6. Be careful with the number of reports you add to a dashboard. If you have too many reports on a dashboard and multiple users are using that dashboard, it could affect overall instance performance.

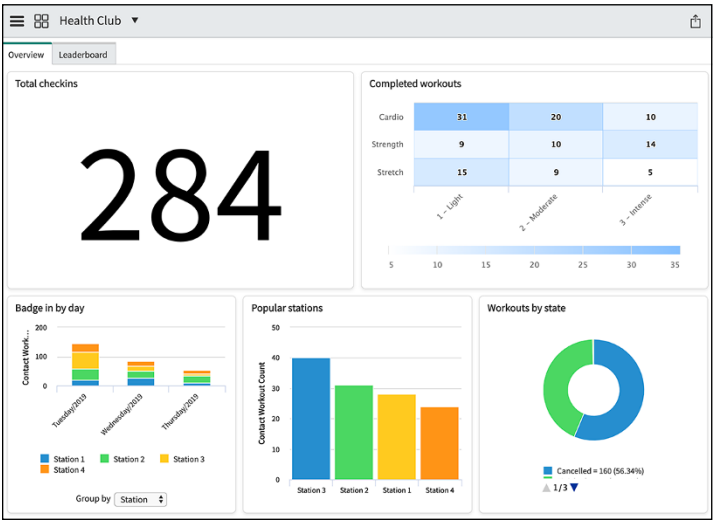


FIGURE 3-6: Dashboards are a useful way to group your reports and gain quick insights.

IN THIS CHAPTER

- Guiding your users with form logic
- Validating user inputs to prevent data issues
- Building more complex process flows
- Integrating your app to external systems
- Sending notifications at key process points

Chapter 4
Logic and Workflow

After you've created your application's data model and provided your users a way to access the data, you're ready to add some logic. Logic is what makes your app a useful tool. It can come in many forms, ranging from *workflow logic* to *form logic* (what people can and can't see or use on a form) to *business logic* (rules that govern what happens to data after it's entered) to *notifications* (making users aware of conditions and events within the app).

Building Dynamic Form Logic

Controlling what users see when they visit a form can greatly increase productivity and responsiveness. For example, users should only see fields that are useful to them — and they may need to see different fields based on what they've selected so far. Several options exist for controlling what's visible, read-only, and mandatory on a form, as well as showing conditional messaging.

To help you decide when to control user access to information, ask yourself the following question: Is this a suggestion or enforcement? A *suggestion* makes the form easier to complete, whereas

enforcement forces the user to do something in order to complete the form.

REMEMBER

User interface (UI) policies are useful for conditional suggestions like showing and hiding fields or adding field messages based on another field's value, while data policies and business rules are better suited for doing conditional enforcement like making a field mandatory.

Figures 4-1 and 4-2 show an example of a UI policy in action. When the category is set to Big, shown in Figure 4-1, the Due Date field is displayed and mandatory (note the asterisk to the left of the Due Date field).

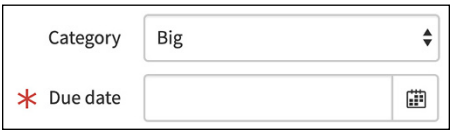


FIGURE 4-1: A UI policy checks the value of the category and displays the due date when the category is Big.

When the category is set to Small, shown in Figure 4-2, the form automatically updates to hide the Due Date field.

FIGURE 4-2: The same UI policy hides the due date when the category is Small.

REMEMBER

The best user experience happens when you utilize both suggestion and enforcement together.

Validating and Simplifying Updates with Business Rules

Business rules are run when a record is created or updated. They're good for building simple conditional logic to run after the form is submitted, like this:

Trigger: *If* this happens on a record,

Action: *then* set this value or show this message.

As an example, let's define a business rule to validate that the due date entered wasn't in the past. The trigger could be that the due date has changed and the date is at or before the current minute. This is easy to construct using the condition builder like the one shown in Figure 4-3.

FIGURE 4-3: Use the condition builder to easily construct business rule triggers.

If the user enters a date in the past, then the trigger condition is true. It displays a message and stops processing as defined on the action part of the form (see Figure 4-4).

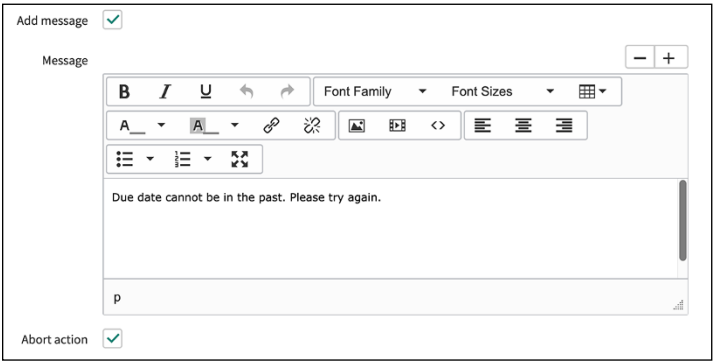


FIGURE 4-4: Update record field values or display a message and stop processing.

More complex logic with multiple steps can be done via Flow Designer.

Controlling Your App with Flow Designer

TIP

Flow Designer allows you to build powerful business workflows. When designing a flow, keep in mind the following tips:

- Each flow should have a singular goal.
- Use sub-flows to create reusable components in a flow (approval is a great example).
- The layout of your flow should clearly indicate its purpose. If there's confusion, consider adding annotations (comments) to the actions.

Start with a whiteboard design of your business flow. Then build the flow, action by action, to align with your process. You may need more than one flow for a single process to stick to these tips.

A basic flow consists of a trigger with one or more actions and logic; it may also include one or more sub-flows. The trigger tells the flow when to start. Flows can be triggered in one of several ways:

- **Record created, updated, or both:** A record on the designated table has been created, updated, or both. You may have certain requirements where you want your flow to trigger on specific conditions for new or updated records. For example, only start an approval on an expense report when the state changes to approval. A condition can be applied to the trigger to filter which record actions can trigger the flow.
- **Incoming email:** You can also run your business logic based on an incoming email from a user or system. For example, when a user sends a new email, create a new record in a table to capture that issue or request.
- **Scheduled:** Run a flow once or on a repeating interval. One example is to find all requested approvals that haven't been updated in the last week and send a reminder notification to the approvers.

- **REST:** You may need to trigger some business logic based on an external trigger such as your customer management system identifying that a new customer has been created. The other system can use REST web service to trigger a flow in ServiceNow.
- **Application:** Application triggers may be associated to a Service Catalog item (for example, ordering a laptop), incoming email, or other triggers. This provides the same easy-to-use Flow Designer capabilities to carry out approvals, notifications, decisions, and more. See how easy automation can be?

Flow Designer actions are the part of the flow that *do* something (for example, send an email notification, update a record, look up records, or create new records). Flow Logic can be applied to make decisions about the data in your flow. There are several logic choices, including "if" (see Figure 4-5) and "decision tree" to conditionally determine whether to run a set of actions (or not), looping constructs like "for-each" and "do-until" to iterate on a list of items like records from a lookup action, or just tell your flow to hang out and wait a certain amount of time.

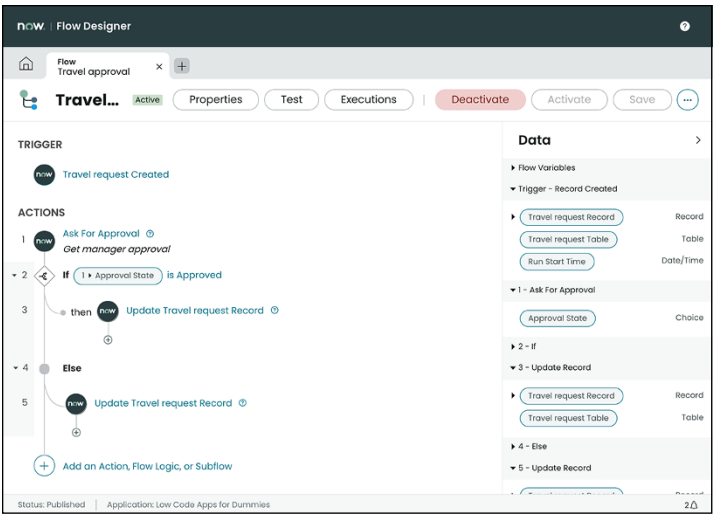


FIGURE 4-5: A simple approval flow in Flow Designer.

Sub-flows allow you to create reusable blocks of actions. For example, say your flow automatically approves and updates a record if the amount is less than $1,000 but requires manager approval for amounts greater than $1,000. You're going to need to do the same update twice in that flow: once when the system auto-approves and again if the manager approves. Why create two sets of the same actions when you can create a sub-flow containing the approval actions and drop it in twice? You've made your flow easier to read and easier to maintain. Experience has taught us that requirements change. When someone asks you to update the approval action, you only have to update it once instead of twice because you've isolated that part of the logic in a sub-flow.

Check out Flow Designer on the ServiceNow Product documentation site at `devlink.sn/fd-docs`, and look at some example videos on the ServiceNow YouTube channel by visiting `devlink.sn/fd-videos`.

Connecting to Third-Party Systems with IntegrationHub

If your app needs to send or receive information to a third-party system, you're going to need an integration. Fortunately, you can use Flow Designer by using prebuilt integration actions from IntegrationHub. In Flow Designer, you simply select from available IntegrationHub bundles of actions called *spokes* (see Figure 4-6) while building a flow to easily allow ServiceNow to interact with other systems in your application landscape. For example, if your team uses Slack for collaboration, you can have your flow send a notification automatically to a Slack channel. For the available spokes in your organization, contact your ServiceNow System Administrator.

REMEMBER

Additional spokes are available from the ServiceNow store at `devlink.sn/spokes-store`. People with some coding and integration skills can also build new spokes. If a particular integration isn't available from ServiceNow, check with ServiceNow to see if

it is on the roadmap. If not, this may be an opportunity to work with your organization's professional developers to build a custom, reusable spoke for that particular system.

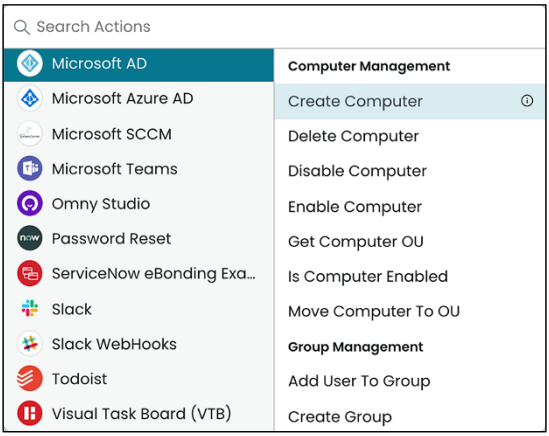


FIGURE 4-6: Connecting to third-party systems is done using IntegrationHub.

Using Notifications to Communicate

Most applications need some sort of email notifications configured. Some examples of that are

- When a task is assigned to a user or group
- When a request is opened or closed on behalf of someone
- When an approval is needed from someone
- At a certain point in a flow

Configuring notifications is quite easy. You only need to identify when to send the notification, who will receive the notification, and what the notification will contain (see Figure 4-7).

The notifications table contains many examples. A quick way to build a new notification is to copy an existing record and change it to suit your purposes.

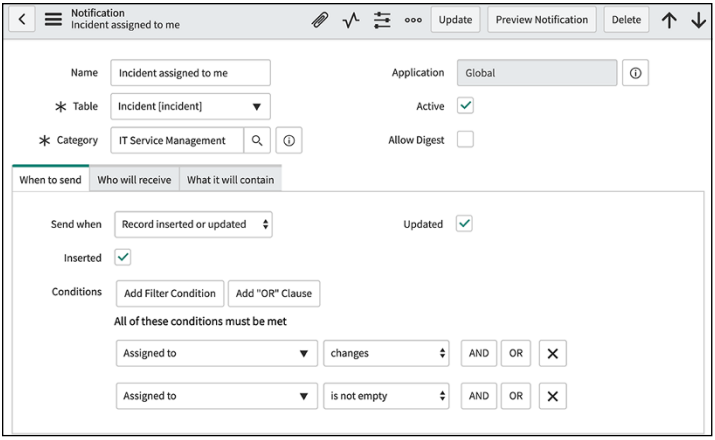


FIGURE 4-7: Use notifications to keep others informed at key points in your process.

TIP

When you're working with notifications, keep in mind the following tips:

- **Consider keeping the Send to Event Creator check box unchecked.** In this case, ServiceNow won't send an email to the person who took the action causing the email to be sent. For example, if you assigned a task to yourself, you don't need to be notified about it.
- **Use email templates if you think you'll be sending out multiple notifications containing the same subject and/or body.** For example, if you're sending an email when a task is assigned to a group, it'll probably contain the same text as an email getting sent to the assigned user, even though the conditions and recipients will be different for both notifications.
- **Use the notification record field labeled Users/Groups in Fields to automatically use the data in the user or group reference field from your app's data record.** Follow this tip instead of specifying a specific user or group (also known as *hardcoding*) in a notification. This suggestion is also one more reason to use a reference field over a string field in your table.

IN THIS CHAPTER

- Adding a chatbot
- Testing your app
- Using surveys
- Offering online tours
- Adding intelligence

Chapter 5
More Low-Code Capabilities

In this chapter, you explore some additional low-code capabilities of ServiceNow that you can use to enhance your app and the user experience. You aren't required to use any of these capabilities in your app, but it's great to know what they are and the value they can add.

REMEMBER

The capabilities mentioned in this chapter are accessed from the standard platform menus, not directly through App Engine Studio (AES) or Studio.

Building a Chatbot

If your app has one or more high-volume/low-complexity tasks, you may want to consider a chatbot. Virtual Agent is a conversational bot platform for providing user assistance through conversations within a messaging interface. Use Virtual Agent to build bots and design bot conversations that help your users quickly obtain information, make decisions, and perform common work tasks.

Components of Virtual Agent

The Virtual Agent platform includes the following components:

- **Virtual Agent conversational (client) interface:** With Virtual Agent, your users interact with a chatbot or live agent through various messaging services. Your users can use the web-based Virtual Agent interface available for Service Portal, Apple iOS, and Google Android environments. They can use the Virtual Agent interface for third-party messaging applications through the ServiceNow integrations for Slack, Microsoft Teams, and Workplace by Facebook.
- **Virtual Agent Designer:** Use Virtual Agent Designer to develop, test, and deploy bot conversations that assist your users with common issues or self-service tasks. Virtual Agent Designer is a graphical tool for building the dialog flows of bot conversations, called *topics.* A topic defines the dialog exchanged between a virtual agent and a user to accomplish a specific goal or resolve an issue.

  Predefined topics are available for Customer Service Management (CSM), HR Service Delivery, and IT Service Management.
- **Live agent handoff:** Give users the option to switch to a human agent for assistance during bot conversations. Virtual Agent is integrated with live chat to offer a seamless transfer from a bot conversation to a live agent. With live chat, you specify the agent chat queues to be used, including the chat interactions transferred from a virtual agent to a human agent. Your users can request a live agent transfer at any time during a chatbot conversation. You can also initiate a live agent transfer through custom conversation flows that you build. You can see an example of this component in Figure 5-1.

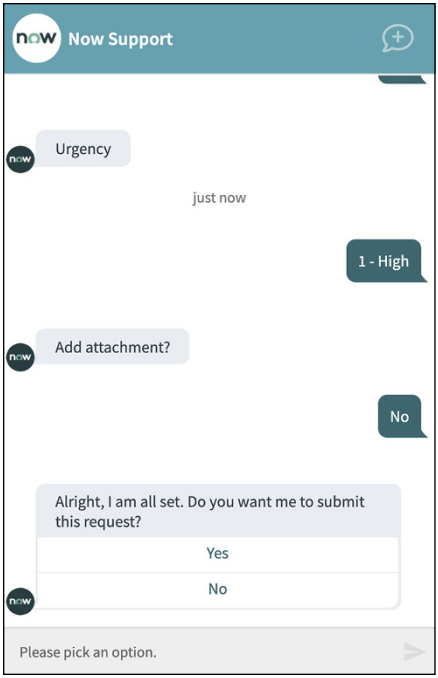


FIGURE 5-1: You can handoff to a live agent as needed with a chatbot.

Benefits of Virtual Agent

Implementing a virtual agent to handle common requests and tasks enables your users to get immediate help, day or night. Providing your virtual agent on channels familiar to your users, such as third-party messaging apps, offers a convenient way for them to get work done quickly. A virtual agent can also offer personalized customer experiences by applying and remembering user information during the conversation.

Typical Tier 2 support tasks that can be accomplished with virtual agents include

- Answering frequently asked questions
- Providing tutorial ("how to") information
- Querying or updating records (for example, getting the current status of cases or incidents)
- Gathering data, such as attachments, for the live agent

- Performing diagnostics
- Resolving multistep problems

Automating these support tasks with a virtual agent frees your support agents to focus on more complex user issues and enables you to scale your support organization accordingly.

You can access Virtual Agent Designer in ServiceNow by choosing Collaboration➪Virtual Agent➪Designer. Take a look and see if your process, users, and app can benefit from a conversation with a chatbot.

Testing Your App

Before deploying your app to production, your IT or app development group may require you to build a test to ensure its functionality and avoid surprises down the road. Even if they don't require a test, it makes sense to invest a few minutes to validate your app.

The Automated Test Framework (ATF) enables you to create and run automated tests to confirm that your instance works after making a change. The change could be a ServiceNow change (something ServiceNow does) or an application configuration change (something you do). If everything tests positive (see Figure 5-2), you can make the changes to production with confidence. If you run in to issues, use the test results to identify changes needing review. You can find ATF under the Automated Test Framework menu.

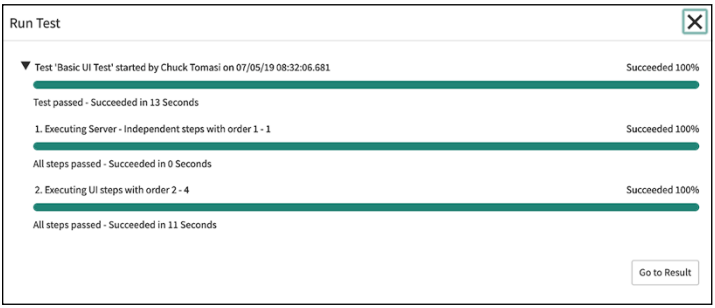


FIGURE 5-2: Validate a release and speed up upgrades with the click of a button with ATF.

Components of ATF

Understanding the components of ATF can help you build more effective and easy-to-run tests. A *test* is made up of steps you define. These include things like opening a form, filling in fields, and validating results. You can run each test individually or as a collection called a *test suite.* Test suites are typically grouped together functionally. For example, if your app is fairly simple, you could have a test suite for your entire app and test it with one click from the test suite.

Benefits of ATF

ATF provides the following benefits for change managers and builders/developers:

- Reduce upgrade and development time by replacing manual testing with automated testing.
- Design tests once and reuse them in different contexts and with different test data sets.
- Keep test instances clean by rolling back test data and changes made after each test runs.
- Create test suites to organize and run tests in batches.
- Schedule test suite runs.
- Enable non-technical test designers to create tests of standard Now Platform functionality.
- Reduce test design time by copying quick start tests and test suites.
- Create custom test steps to expand test coverage.

We like to say that ATF tests are the gift that keeps on giving. Once defined, they can be used to validate your app releases and also have a bonus when it comes to upgrading your instances. Your tests are already defined and ready to go. At the push of a button, you can validate your app quickly and get the value of ServiceNow's latest releases faster.

Sending Surveys

With the ServiceNow Survey Management application, shown in Figure 5-3, you can create, send, and collect responses for basic surveys. The survey designer lets you create survey categories and questions, configure the details, and publish the survey to specific users or groups.

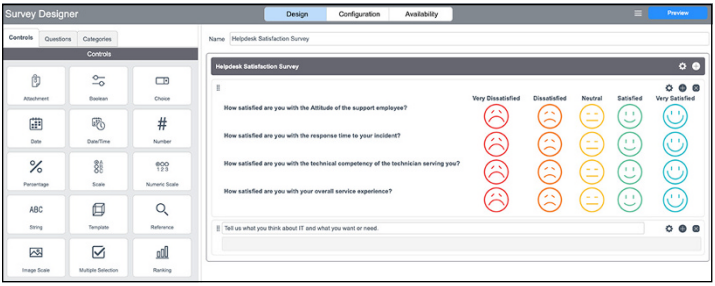


FIGURE 5-3: Create and configure surveys using the Survey Designer.

You can assign a survey to individual users or groups who receive all the questions from all the categories. You can also customize each question and make it dependent on the response to another question. The following describes the procedure to create and publish a survey.

1. **Create survey categories.**
2. **Create questions within each category.**
3. **Configure survey details, such as introductory and closing remarks and time limit.**
4. **Select recipients for the survey.**
5. **Publish the survey to the selected users or groups.**

Surveys can be a useful metric to determine if your app is successful. You can find surveys in the ServiceNow app under the Survey menu on the right navigation menu. Check out Figure 5-4.

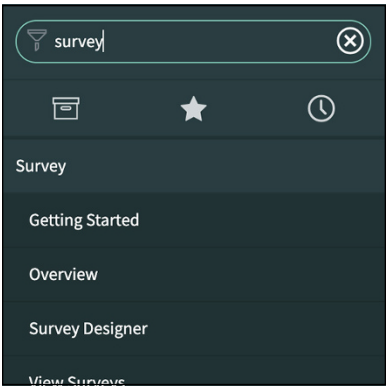


FIGURE 5-4: The Survey application menu (partial shown).

Offering Self-Paced, Onscreen Training

Guided tours help train and onboard users within the ServiceNow user interface (UI). Each tour contains a series of interactive steps that help users complete online tasks within the browser window. Administrators can create tours for ServiceNow applications, service portals, and custom applications. Figure 5-5 shows you the Guided Tour Designer that's used to create tours that demonstrate how to use a feature. For example, you can create a tour to represent a training model for specific policies and processes, such as creating a new claim or reviewing expense reports.

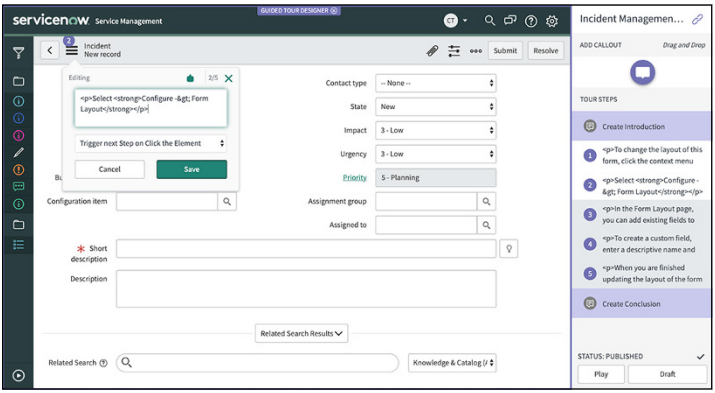


FIGURE 5-5: The Guided Tour Designer.

Guided tours use a series of steps that may span multiple pages. You can create purely informational steps that users read through and acknowledge, which results in no change to the ServiceNow instance. Alternatively, you can provide users with an interactive experience where they click through and actively work with the application at hand. For example, an Introduction to Incidents tour may simply show them the key features of the Incidents table, while a Create Your First Incident tour would walk them through creating a real incident, which results in a new record in the Incidents list.

A guided tour can accelerate adoption of your app and reduce supplemental training and documentation. Find guided tours under Guided Tour Designer on the left navigation menu of the ServiceNow app, as shown in Figure 5-6.

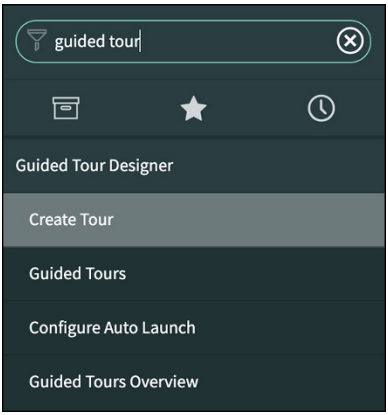


FIGURE 5-6: The Guided Tour application menu.

Adding Intelligence

The Now Intelligence products can help you lower costs and increase productivity through process improvement, self-service, and automation. Service owners can deliver and refine artificial intelligence (AI) capabilities quickly, gaining greater insight into real-time patterns and trends for service delivery teams. This information enables you to make better, faster decisions — without the need for data science expertise.

Requesting help

If you want to provide your users with a way to interact with your applications using natural language, you can use the built-in technology called *Natural Language Understanding* (NLU).

Say you've written a time-off request app so your employees can request some well-deserved vacation. By using the Virtual Agent chatbot technology, you can easily define a model to allow employees to interface with the chatbot using statements such as "I'm requesting a week off starting next Monday." Based on your model, the system can interpret the intent of the statement as your employee's existing Leave of Absence request, a start date of June 21, 2021, and the duration as a week. That's enough information to submit the request.

Requesting information

To heighten the NLU experience (see the preceding section), you can use Natural Language Query (NLQ). Along with making it easy to enter information, Now Intelligence products can also retrieve or report information with NLQ. Reduce the number of requests you get to create specific reports by enabling users to get their information with human queries. For example, typing "open p1 incidents for my team" returns a list of incidents where the priority is 1, and the records are assigned to any member of your team. NLQ is available within reporting or from the standard platform lists, as shown in Figure 5-7.

FIGURE 5-7: Clicking the speech icon at the top of a list activates the NLQ prompt.

Performance Analytics

ServiceNow Performance Analytics (PA) is an in-platform process optimization solution to create management dashboards, report on key performance indicators (KPIs) and metrics, and answer key business questions to help increase quality and reduce the costs of service delivery.

PA enables businesses to set, track, and analyze progress against goals. The products help you improve performance and accelerate continual service improvement by

- Tracking critical process metrics and trends
- Measuring process health and behavior against organizational targets
- Identifying process patterns and potential bottlenecks before they occur
- Continually visualizing the health of processes through both historical and real-time statistics in role-based dashboards, so you and your business can make informed decisions

IN THIS CHAPTER

- Planning effectively
- Creating your tables and fields
- Keeping your app running smoothly
- Testing your apps
- Partnering with developers

Chapter 6
Ten Tips for Low-Code App Development

The tips in this chapter come from years of experience across hundreds of implementations by customers, partners, and ServiceNow developers. We hope that following these ten tips when building your app brings you much success!

Make a Plan

When you begin with a plan, you have a clear picture of the outcome you're looking for and how to get there, and that means you have a greater chance of success. For more info on this tip, check out Chapter 1.

Name Tables and Fields

REMEMBER

As you build the tables and fields that make up your data model, remember to consider good naming of tables and fields. Labels can be changed later, but names can't. Flip back to Chapter 2 for more information.

Consider Common Personas and Roles

In most apps, there are several default roles to consider when you build your app. A user (or requester) role is often assigned to someone who typically creates the record, checks the status, and makes small updates on his own behalf. An admin is someone responsible for the management and maintenance of the app. We often see a role for someone responsible for interacting with the record to drive the process; this persona may have a name like agent or fulfiller. You may opt to create a separate role for an approver, who's only involved with a specific approval step in your app. Check out Chapter 2 for more information.

Use Good Form and List Layout

Consider the experience users will have as they interact with your app. Watch how they move through your app. Are there common things they do and in a specific order? Can you adjust the layout to make it easier for them to navigate and perhaps save time? See Chapter 3 for more information on paying attention to the user experience.

Take Advantage of Different Field Types

ServiceNow offers several different field types. Take a look at the options available and consider how they may make your app more effective. You may find that changing a date and time field to a date gives you better reporting capabilities or using a URL field instead of a text field allows users to click a link and be taken to that site. Head to Chapter 2 for more on field types.

Avoid Deleting Records

Generally, deleting records is bad because it can lead to data inconsistency issues. A better method is to add a True/False field (often called *Active*) to enable you to deactivate records like those of former employees. After that, you can use a filter such as "Active | is | true" to display just the active records. This is also known as a *reference qualifier*.

Test Your App

Use the Automated Test Framework (ATF) to create tests for your app. Remember to use both good test cases (where you expect it to pass) and bad test cases (where you expect it to fail). When you use ATF, you can validate changes before going to production and reduce upgrade time between ServiceNow versions. We cover ATF in detail in Chapter 5.

Get Familiar with the Commonly Used Tables

Understanding the built-in tables can save you development time, reduce the need for integrations, and improve cross departmental workflow. In Chapter 2, you can review the list of commonly used tables.

Limit the Number of Records Retrieved in a Report

More data equals more time. If you find your dashboards, reports, or lists are taking an uncomfortable amount of time to display, consider adding filters such as "created | on | today" or "assigned to | is (dynamic) | me" to reduce the amount of data retrieved (see Chapter 3).

Work with Your Developers

ServiceNow is a single development platform that accommodates developers of all skill levels working cooperatively. This helps more people to build using low-code capabilities and to leverage the developer skill set as needed.

TIP

If you find your app needs complex logic or custom integrations that go beyond the low-code capabilities, work with the developers in your organization or seek out an implementation partner. You can reap two specific benefits:

- Less backlog for the developers
- Faster time to value for you

Notes

Notes

Notes

Notes

Notes

Notes

Start building apps fast

In pursuit of a better way to work, many organizations have a priority to digitize their processes. The best way to do that is to capture those processes within smart workflow apps that can streamline and automate those processes for improved experiences and greater productivity. This book helps you learn how you can quickly build your own apps, regardless of skill or experience level, so you can power your own digital transformation at scale.

Inside...

- Discover what types of apps you can build
- Learn best practices from the experts
- Follow a proven development plan
- Explore many low-code capabilities
- Unlock productivity in your organization
- Create amazing experiences
- Give others insight to your processes

servicenow

Chuck Tomasi is a Sr. Developer Advocate at ServiceNow with nearly 40 years of industry experience. A leader in the ServiceNow community, he's produced thousands of podcasts and videos. **Brad Tilton** is a Developer Advocate at ServiceNow who's produced videos, blog posts, best practice guides, and workshops around building apps in ServiceNow.

ISBN: 978-1-119-82038-3
Not For Resale

for dummies®
A Wiley Brand

Also available as an e-book

WILEY END USER LICENSE AGREEMENT

Go to www.wiley.com/go/eula to access Wiley’s ebook EULA.