Use industry templates for advanced case management, Part 2: Introducing the Auto Claims Management sample solution template for IBM Case Manager

Building an auto claims solution with IBM Case Manager

Skill Level: Intermediate

Jos H. Olminkhof (jolminkh@us.ibm.com)
ECM Industry Solutions
IBM

20 Jan 2011

Updated 01 Mar 2012

IBM® Case Manager provides the platform and tools for the business analyst to define and implement a new generation of case management solutions. To accelerate the development of solutions in particular industries, IBM Case Manager supports the notion of a solution template: a collection of case management assets that can be customized and extended to build a complete solution. To help illustrate the value of solution templates and the abilities of IBM Case Manager, IBM has provided two sample solution templates that can be used as learning tools for new users of the platform. This tutorial introduces one of those templates — Auto Claims Management — from the insurance services industry. Gain an understanding of what a template is, and learn about the assets delivered in this sample template and how they were built. (This tutorial includes the code for this sample template, as well as instructions on how to deploy it.)

Section 1. Before you start

Introduction

Case management is a critical function for virtually every business. The ability to efficiently process cases has impact on the top and bottom lines, could have regulatory implications, and most importantly, is critical to maintaining customer satisfaction and loyalty. Effectively implementing a case management solution that meets today's requirements requires a broad platform that brings together a range of capabilities, including content management, business process management, business rules, collaboration, and analytics. The solution needs to integrate seamlessly with existing infrastructure, leveraging those assets to provide complete, consistent context to case workers in a user interface tailored to how and where they work.

IBM Case Manager provides the platform and tools for the business analyst to define and implement this new generation of case management solutions. To accelerate the development of solutions in particular industries, IBM Case Manager supports the notion of a solution template, which is a collection of case management assets that can be customized and extended to build a complete solution. To help illustrate the value of solution templates and the abilities of IBM Case Manager, IBM has provided two sample solution templates that can be used as learning tools for new users of the platform. This tutorial introduces one of those templates — Auto Claims Management — from the insurance services industry. After reading this tutorial, you will have an understanding of what a template is, and you will understand the assets delivered in this sample template and how they were built. (This tutorial includes the code for this sample template, as well as instructions on how to deploy it.)

Prerequisites

This tutorial is intended to supplement your knowledge of IBM Case Manager by providing a deeper understanding of how to build and deploy a solution. As a result, this tutorial assumes at least a beginning level of familiarity with IBM Case Manager and the IBM Case Manager Builder, as well as the IBM FileNet® P8 Platform and tools. Readers should have IBM Case Manager installed and have some experience working with the tools to build a case management solution. For more information about IBM Case Manager, including a guided tour, see Resources.

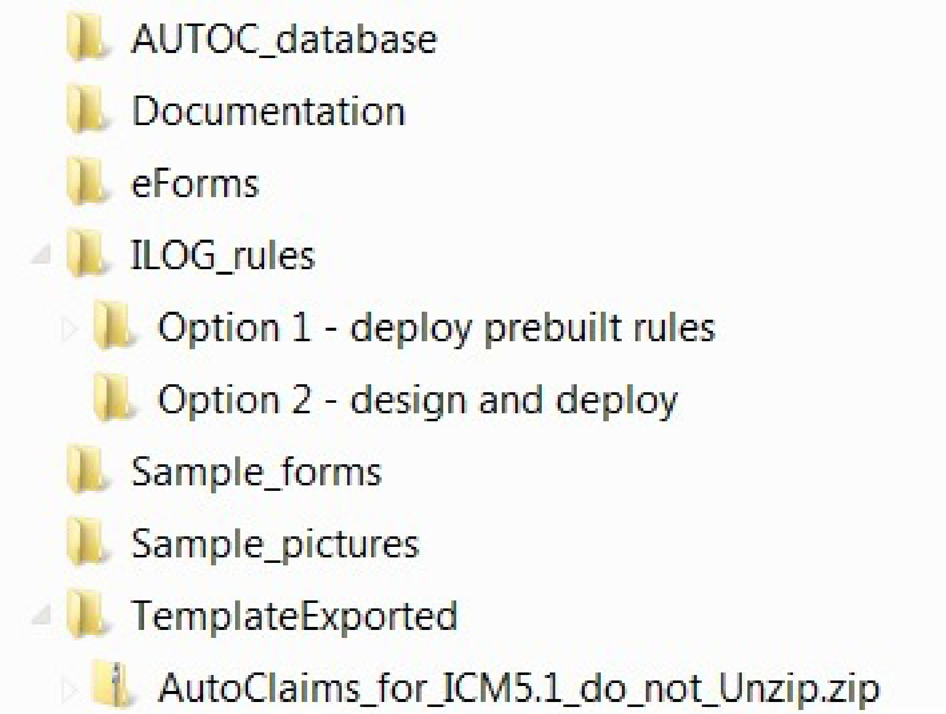
Target audience

This tutorial is targeted at business analysts and information technology personnel looking to supplement their understanding of IBM Case Manager, eForm authoring, and their use of business rules with an understanding of templates and some concrete examples of building particular assets.

Deploying the sample template

Included with this tutorial is a ZIP file containing the artifacts of the solution (see Download). When you are ready to deploy the solution, open the file and extract the contents onto your local file system. You will see the following directory structure:

Figure 1. Contents of ZIP file



- AUTOC_database
- Documentation
- eForms
- ILOG_rules
  - Option 1 - deploy prebuilt rules
  - Option 2 - design and deploy
- Sample_forms
- Sample_pictures
- TemplateExported
  - AutoClaims_for_ICM5.1_do_not_Unzip.zip

The Documentation folder contains a document named Deploying the Auto Claim Management Template - IBM Case Manager 5.1.pdf. This document contains the detailed instructions for deploying the sample template, creating a solution from it, deploying the solution, and running it. The folder also contains a demo script that will guide you through the solution and help you become familiar with the assets.

The steps to deploy the template and then create a solution from it are straightforward. It should take you no more than a couple of hours to deploy the template. While it is not a prerequisite to reading the tutorial, you may wish to deploy the solution before you read the tutorial so that you can directly reference the code as particular topics are discussed.

Section 2. Introduction to IBM Case Manager and the

solution templates

Overview of IBM Case Manager

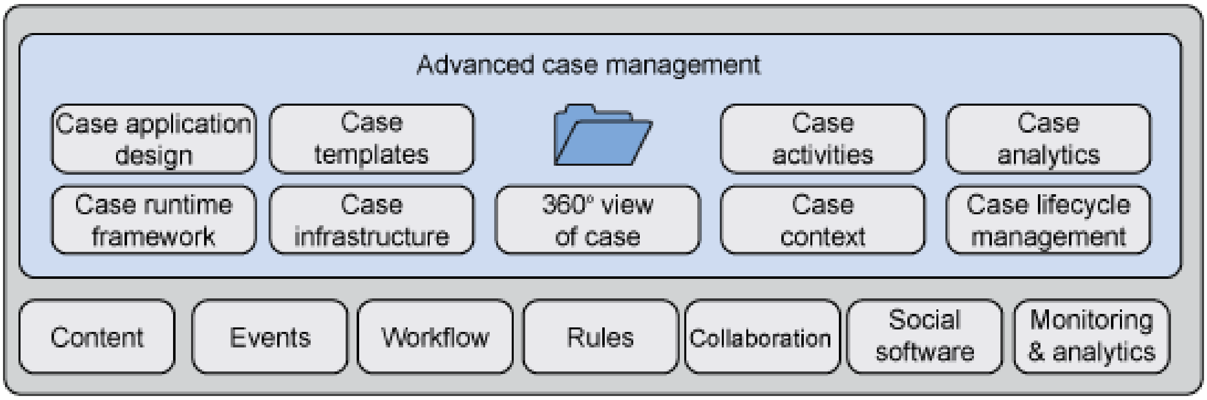
There are many types of case management applications. For the purposes of this tutorial, let's focus on case management solutions that are more human-centric in nature and require the coordination of different services throughout an organization, such as health, legal, or financial, in order to provide a business service for a customer or party. Typically, this includes creating a case file and following a process (or collection of processes) to ensure the delivery of the business service. Case-related information is used by case managers or members of a case team who collaborate to resolve and close a case. After the case is completed, information about the case is typically retained for compliance or in support of longer-term business processes.

These case management scenarios require highly collaborative, dynamic, and event-driven work. Often, cases and the processes that support them can have significant life cycles, and can be closed, suspended, and reopened.

An example case management solution is auto claims management in insurance services. In this scenario, a customer calls in to file a claim, and a case is opened to track this claim. Resolving the claim requires collaborative work between multiple teams, inside and outside of the organization. There are regulatory and company service-level agreements (SLAs) that dictate timelines and interactions with the customer.

Delivering this type of solution requires the effective integration and leveraging of a broad set of capabilities as illustrated in Figure 2.

Figure 2. Artifacts in advanced case management



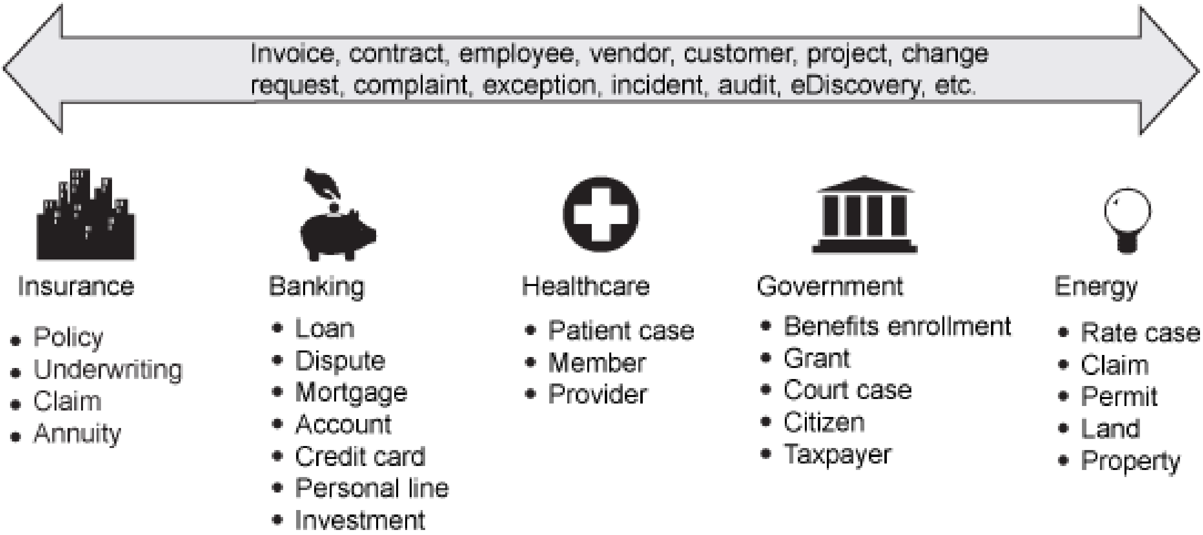
Case content must be gathered and maintained in a single, secure location and passed seamlessly between all individuals working on the case. Context must be complete and consistent across all users. Events, business rules, and workflow help to automate elements of the process to reduce errors, ease the burden on the user, and increase efficiency. Efficient collaboration between team members is critical. Providing transparency into the process is important, not only for operational purposes but also for assessing the impact of the solution on the overall business.

IBM Case Manager provides a platform that leverages capabilities from across IBM Software Group to deliver the advanced requirements needed to drive better case outcomes. The platform combines enterprise content management and business process management, along with integrated rules, events, collaboration, and analytics to deliver a comprehensive case management product. On top of this platform, IBM Case Manager provides a set of tooling that enables business analysts to quickly define solutions, and then collaborate with customers and IT to deliver them.

Case management solution templates

There are case management solutions in virtually every industry. Figure 3 illustrates some of these case management solutions.

Figure 3. Different types of cases across industries



Most of these case types are vertical in nature, solving a specific problem in a specific industry. But some are more horizontal, addressing a challenge that occurs in many industries.

To aid in the deployment and delivery of these solutions, IBM Case Manager supports the notion of a *solution template.* A solution template contains case management assets that are applicable to a particular business problem, and can be easily customized and extended to deliver a complete solution.

To illustrate the concept of a template and demonstrate how the features of IBM Case Manager can be used to deliver a solution, IBM has provided two sample solution templates coincident with the first release of the product. The first sample is Credit Card Dispute Management in financial services. This template is discussed in Part 1 of this series.

The second template is Auto Claims Management in the insurance industry. The remainder of this tutorial introduces this template and the assets it contains, and describes some details of how the template was built. Finally, this tutorial shows you how to create a template and how you can deploy the Auto Claims Management template in your environment.

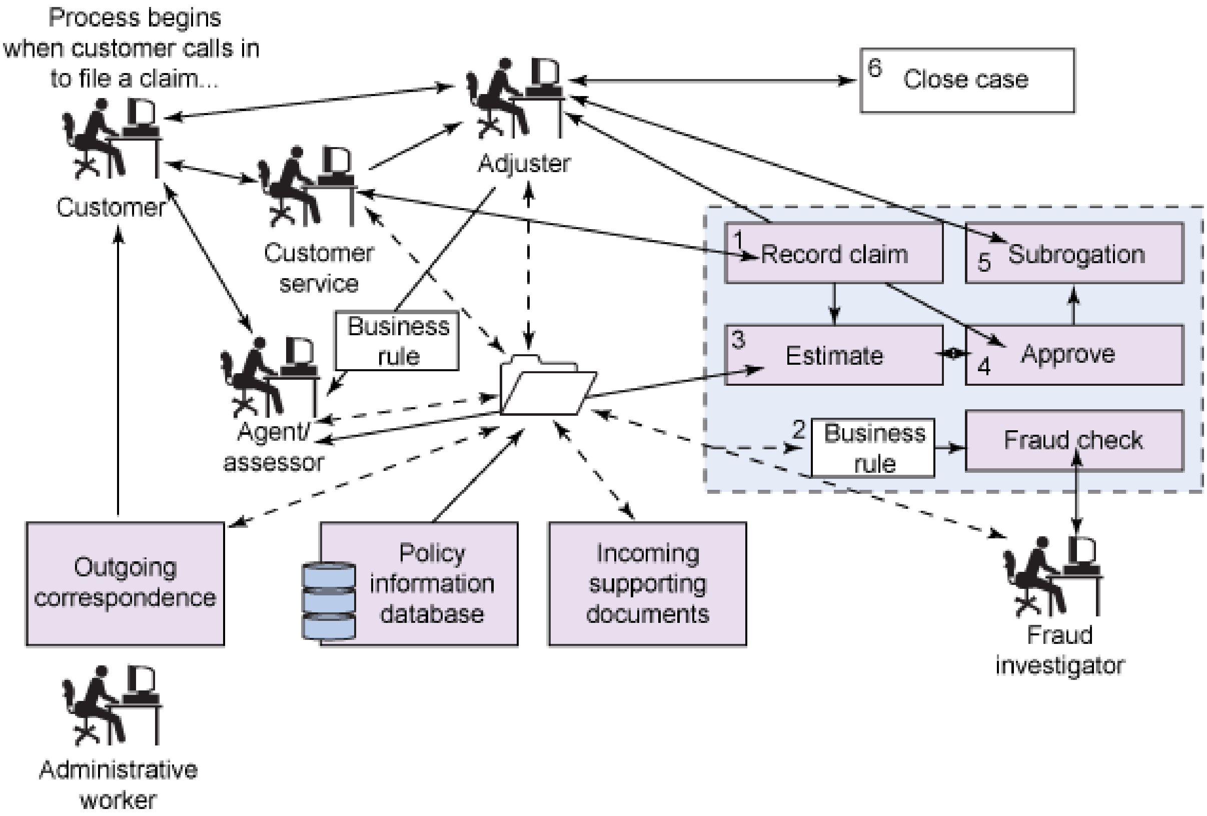
Auto claims management

Successfully managing auto claims requires efficient interactions among customers, damage assessors, medical and legal experts, and the insurance company. The expertise required for a claim will vary, depending on the details of the claim. Most insurance companies have well-defined processes for auto claims. Errors or inefficiencies in the interactions among experts, customers, and the insurance company can result in additional costs, delays, and unresolved claims disputes.

IBM Case Manager provides capabilities that allow collaboration and flexibility in the business process, while recording the events. The Auto Claims Sample Solution Template is a good starting point for building your Auto Claims solution.

The diagram in Figure 4 provides a high-level view of the case flow.

Figure 4. High-level process flow for managing an auto claim



The process typically starts with a customer involved in an accident calling the insurance company. A customer service representative takes the call and works with the customer to complete a First Notice of Loss (FNOL) form. In addition to details about the customer and the accident, the form will allow the customer service representative to specify additional steps, such as whether a rental car is required, or whether a third party might be responsible for part or all of the damage. If the customer has any supporting documents, such as pictures or the police report, she will send those in, and the customer service representative will include those with the case.

At this point, a claims adjuster takes ownership of the case and works with third-party agents to gather estimates and arrange repairs. She will also approve the claim and work with other insurance companies, if necessary, to pay for the repairs.

Underlying this process are workflows and rules systems that route work and inspect the case data to determine if particular steps need to be taken, such as a fraud investigation.

Section 3. Implementing the Auto Claims Management

template

IBM Case Manager enables the business analyst to design, build, test, and deploy a solution using a tool called IBM Case Manager Builder. The Case Manager Builder provides an easy-to-use environment that allows the analyst to describe the solution using familiar terminology, while abstracting away the underlying complexities of the technical implementation. (Throughout this tutorial, you will see numerous screenshots of the Case Builder tool.)

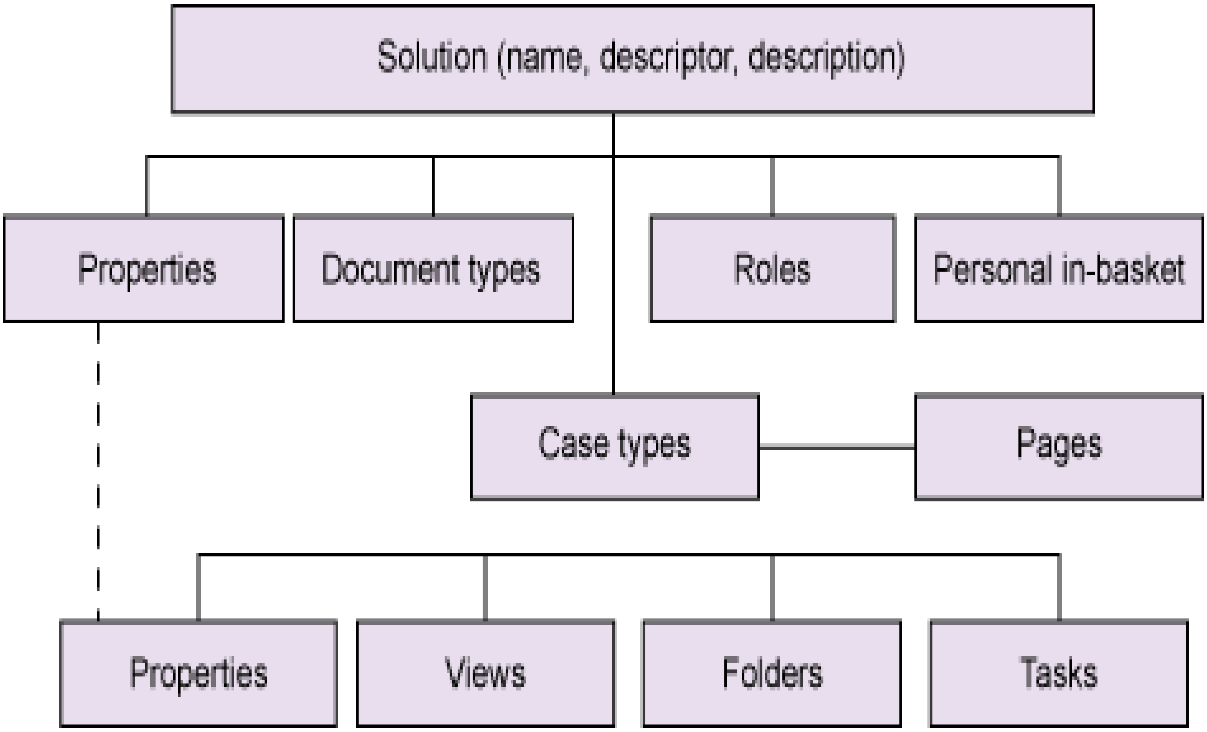
In describing the implementation of the Auto Claims Management sample, this tutorial starts with the assets defined using the Case Builder tool. It then describes some places where we extended the template using the IBM FileNet P8 Process Designer and discusses how we implemented two rules that we built with ILOG® JRules.

Finally, the tutorial inspects some UI assets contained in the template.

The object model

The artifacts of a case management solution are organized into an object model, as illustrated in Figure 5.

Figure 5. The IBM Case Manager object model



Each solution includes a unique, user-provided descriptor that will be used as a prefix on all assets within the solution. The solution also contains one or more properties, document types, roles, and case types. Document types, properties, and roles are defined at the solution level and are used across the defined case types. There is one personal in-basket defined for all users of the solution.

Within a case type are properties, case property views, folders, and tasks. All properties defined at the case-type level are mapped up to the solution level so they can be reused in other case types. Case types also contain user pages. These pages can be customized, and new pages can be created to meet the needs of the case workers.

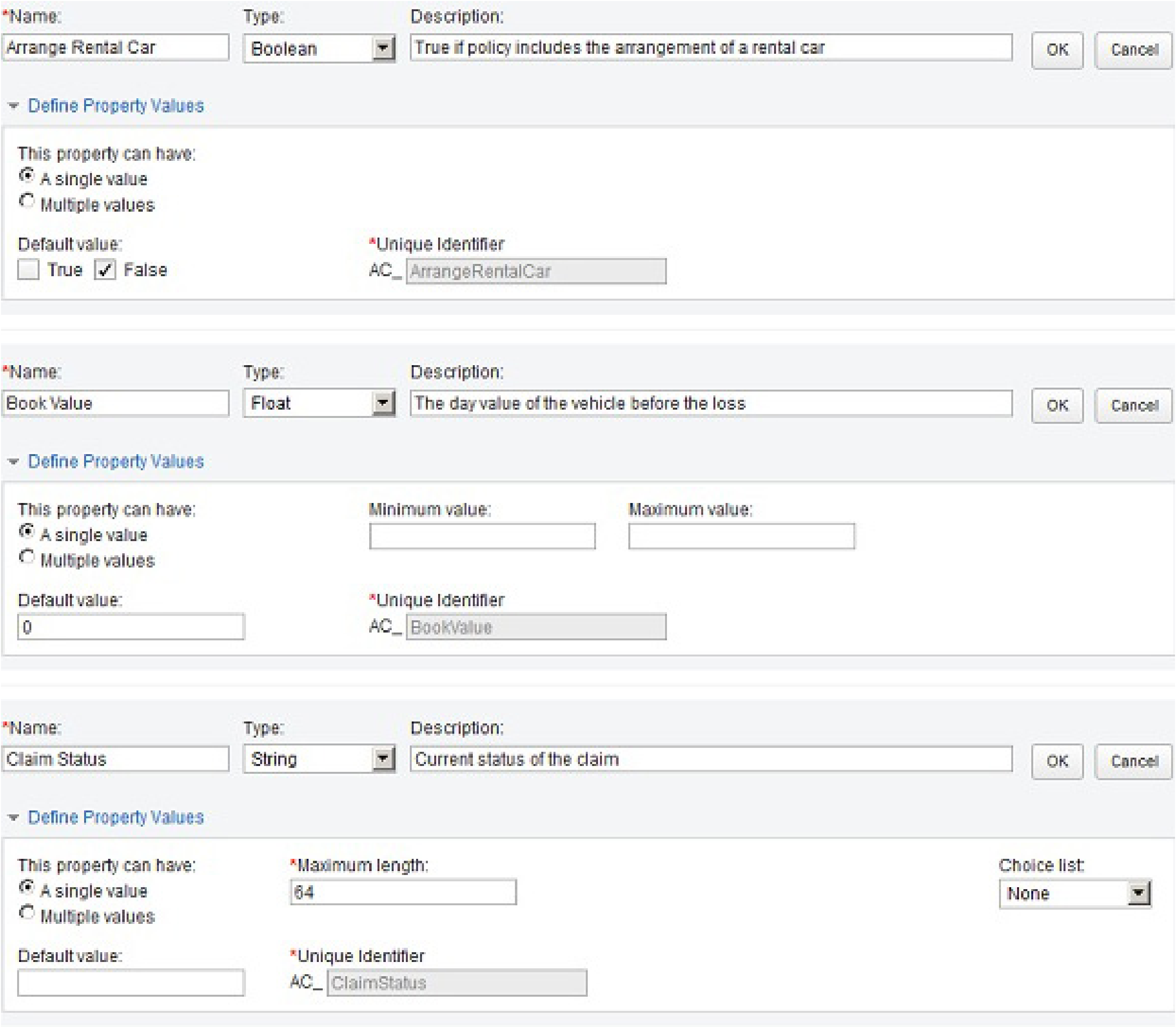
Case Builder artifacts for the Auto Claims sample template

This section discusses all of the assets defined and implemented using the IBM Case Manager Builder tool. Collectively, they provide the definition and high-level structure of the template.

Properties

There are a significant number of properties defined in the template. Most of them are self-explanatory; they store and display details about the customer's policy and the claim being worked. Figure 6 shows the details defined for three types of properties.

Figure 6. Details of three types of properties



*Name: Arrange Rental Car
Type: Boolean
Description: True if policy includes the arrangement of a rental car

Define Property Values

This property can have:
● A single value
○ Multiple values

Default value: □ True ☑ False

*Unique Identifier AC_ ArrangeRentalCar

*Name: Book Value
Type: Float
Description: The day value of the vehicle before the loss

Define Property Values

This property can have:
● A single value
○ Multiple values

Minimum value:
Maximum value:

Default value: 0

*Unique Identifier AC_ BookValue

*Name: Claim Status
Type: String
Description: Current status of the claim

Define Property Values

This property can have:
● A single value
○ Multiple values

*Maximum length: 64

Choice list: None

Default value:

*Unique Identifier AC_ ClaimStatus

Other properties help drive the solution in a variety of ways:

- **Setting a precondition for a task** — Note that a property can be used as a precondition for the instantiation of a new task. In the Auto Claims sample, there are a number of Boolean properties that the case workers can set to start new tasks. When the case worker sets one of them to true, a new work stream is created to complete that piece of work. As an example, when the claims adjuster sets the Boolean property Request Estimate to TRUE, the *create estimate* task will automatically start, since `Request Estimate = TRUE` is a precondition for that task.
- **Specifying the purpose of a work item** — The property Work Item is used to indicate what the purpose of each work item is in a case worker's inbox.

- **Indicating the current state of the case** — The property Claim Status is used to indicate the case's current state.

Roles and in-baskets

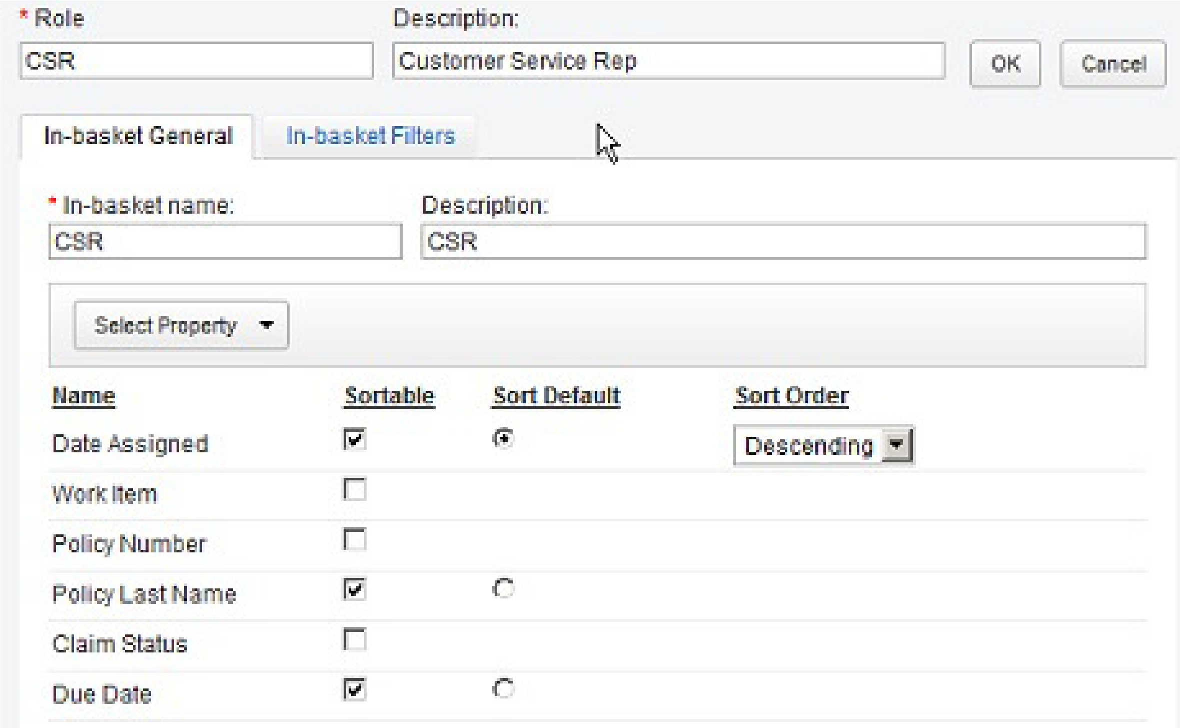
There are a number of roles defined for the solution. See Table 1 for a list of them.

Table 1. Roles used in the Auto Claims template

Role name	Description
CSR	Customer service representative
Adjuster	Claims adjuster
Agent	Third-party investigating/assessing damage
Supervisor	Supervisor for team of adjusters
Recovery Expert	Expert for recovering the cost of losses
Fraud Investigator	Investigates potential fraud cases
Administrative Clerk	Performs back-office activities, like assisting with the generation of letters

As Figure 7 shows, each role has an associated in-basket.

Figure 7. Details of a role



* Role: CSR

Description: Customer Service Rep

* In-basket name: CSR

Description: CSR

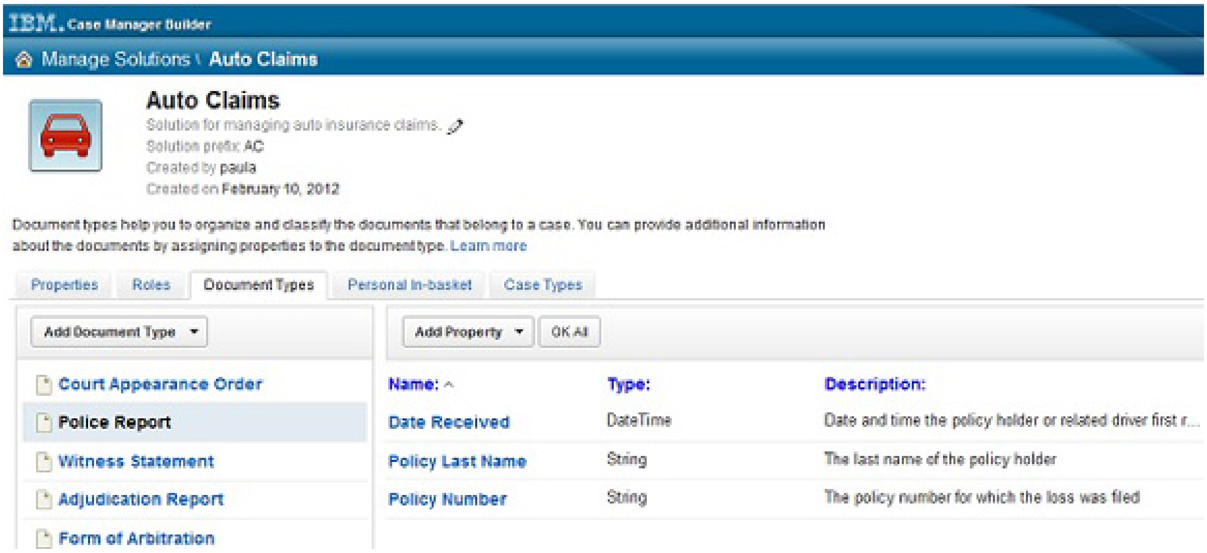
Name	Sortable	Sort Default	Sort Order
Date Assigned	☑	●	Descending
Work Item	□		
Policy Number	□		
Policy Last Name	☑	○	
Claim Status	□		
Due Date	☑	○	

In addition, the template defines a simple personal inbox for handling work items directed at individuals or workgroups.

Document types

Throughout the auto claims process, there are documents that may be included with the case. Figure 8 shows that properties can be assigned to describe the document type.

Figure 8. Details of a document type



Auto Claims

Solution for managing auto insurance claims.

Solution prefix: AC

Created by paula

Created on February 10, 2012

Document types help you to organize and classify the documents that belong to a case. You can provide additional information about the documents by assigning properties to the document type. Learn more

- Court Appearance Order
- Police Report
- Witness Statement
- Adjudication Report
- Form of Arbitration

Name:	Type:	Description:
Date Received	DateTime	Date and time the policy holder or related driver first r...
Policy Last Name	String	The last name of the policy holder
Policy Number	String	The policy number for which the loss was filed

Table 2 lists the document types that categorize these documents.

Table 2. Document types used in the Auto Claims template

Document type	Description/Notes
FNOL	First Notice of Loss (PDF version only)
Police report	
Witness statement	
Adjudication report	Report of the decisions made in the legal adjudication process
Form of arbitration	
Expertise form	A form that an external specialist can provide to indicate his qualifications
Subrogation form	Agreement when multiple parties share liability for a loss
Proposal of liability share	
Piece of evidence	Example: Photos taken at the scene of the loss
Medical report	
Note lawyer	
Correspondence	

Claim form	The claim form that the claimant has downloaded and filled out

Case types

There are three case types defined within the template. The first is *general claim,* which defines the processes for simple claims with material damage only (approximately 80 percent of all claims). This is the case type discussed in detail in this tutorial and that has been implemented with a set of tasks.

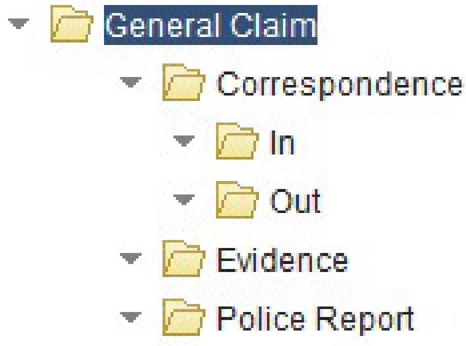
The two other case types, *claim including bodily injury* and *major loss claim,* are provided solely for the purpose to illustrate that the case worker who initiates the case has a choice to select different claim types. IBM Case Manager offers the capability to define multiple case types, which may involve different roles, different tasks, and different UIs for the execution of these tasks.

As an example, when the company receives a bodily injury claim, the claims adjuster will be the central figure in the case and will ask numerous questions during the process. The claims adjuster will be surrounded by legal and medical specialists. There will be hospital and other medical bills (consider the ambulance ride and emergency room visit).

Other types of auto claims that may require a different case type are property damage claims, windshield claims, or third-party insurance claims, where the claimant is not the policy holder, but somebody filing a claim against the policy holder.

The general claim case type case structure

Figure 9. Folder structure for the general claim case type



During the execution of the claims process, documents are being added to the case and organized into a folder structure. For this template, three folders have been created:

- Correspondence — To capture all correspondence with the customer and other third parties.
- Evidence — To capture pictures the customer has taken of the scene of the accident and potential additional evidence material gathered by the agent who assesses and estimates the damage or the claims adjuster.
- Police report — To file the police report.

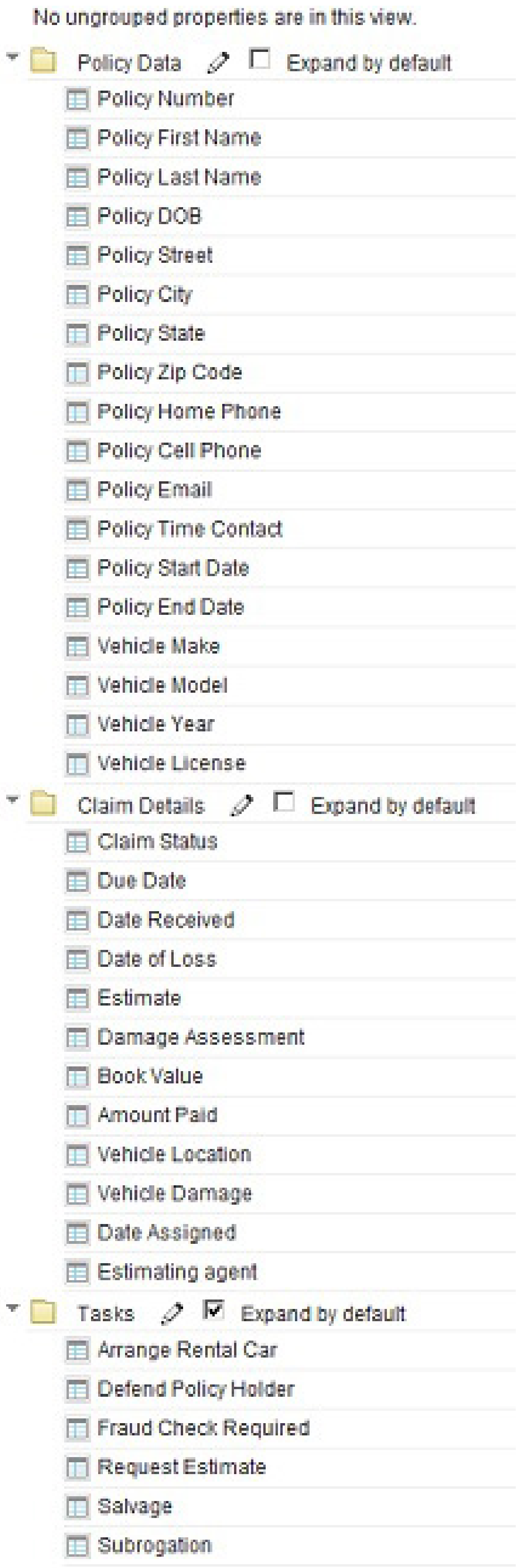
Properties displayed in views

To help organize the case information for the case worker, the Case Builder allows the business analyst to create a view of the properties in the case. For the Auto Claims template, the case information is organized into three groups:

- Policy details, which includes information such as policy number, name, address, cars insured, etc.
- Claim details, which includes properties about the claim itself.
- Tasks work actions that the case worker can initiate as he learns more about the case.

Figure 10 shows the data view with each group expanded.

Figure 10. The views of the properties in the general claim case type



No ungrouped properties are in this view.

Policy Data □ Expand by default

- Policy Number
- Policy First Name
- Policy Last Name
- Policy DOB
- Policy Street
- Policy City
- Policy State
- Policy Zip Code
- Policy Home Phone
- Policy Cell Phone
- Policy Email
- Policy Time Contact
- Policy Start Date
- Policy End Date
- Vehicle Make
- Vehicle Model
- Vehicle Year
- Vehicle License

Claim Details □ Expand by default

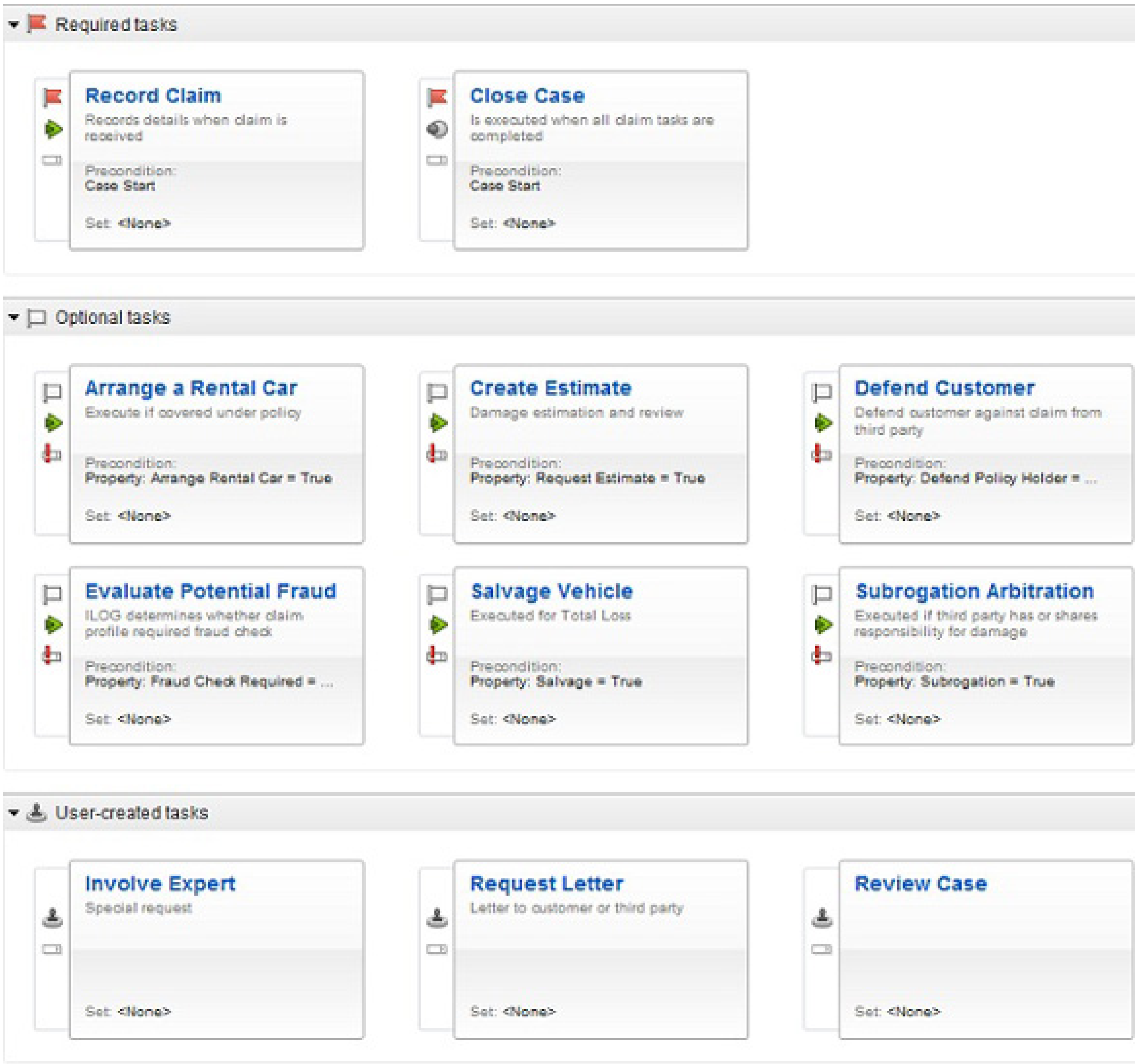
- Claim Status
- Due Date
- Date Received
- Date of Loss
- Estimate
- Damage Assessment
- Book Value
- Amount Paid
- Vehicle Location
- Vehicle Damage
- Date Assigned
- Estimating agent

Tasks ☑ Expand by default

- Arrange Rental Car
- Defend Policy Holder
- Fraud Check Required
- Request Estimate
- Salvage
- Subrogation

The *general claim* case type contains a number of tasks, as shown in Figure 11.

Figure 11. The tasks in the general claim case type



Required tasks

Record Claim
Records details when claim is received
Precondition:
Case Start
Set: <None>

Close Case
Is executed when all claim tasks are completed
Precondition:
Case Start
Set: <None>

Optional tasks

Arrange a Rental Car
Execute if covered under policy
Precondition:
Property: Arrange Rental Car = True
Set: <None>

Create Estimate
Damage estimation and review
Precondition:
Property: Request Estimate = True
Set: <None>

Defend Customer
Defend customer against claim from third party
Precondition:
Property: Defend Policy Holder = ...
Set: <None>

Evaluate Potential Fraud
ILOG determines whether claim profile required fraud check
Precondition:
Property: Fraud Check Required = ...
Set: <None>

Salvage Vehicle
Executed for Total Loss
Precondition:
Property: Salvage = True
Set: <None>

Subrogation Arbitration
Executed if third party has or shares responsibility for damage
Precondition:
Property: Subrogation = True
Set: <None>

User-created tasks

Involve Expert
Special request
Set: <None>

Request Letter
Letter to customer or third party
Set: <None>

Review Case
Set: <None>

For Auto Claims, there are two required tasks: record claim and close case. Within the *record claim* task, the CSR gathers all the information required to process the claim. Depending on the information gathered in the *record claim* task, a number of other optional tasks will be instantiated.

All of the optional tasks in the Auto Claims template are automatic, instantiated when certain preconditions are met. The preconditions can be set programmatically in a workflow or by a background process, such as an ILOG rules system. They can also be set manually by the case worker.

There are two user-created tasks defined in the template. *Request letter* enables a case worker to request a particular type of letter be sent. *Involve recovery expert* would be invoked by the claims adjuster when additional assistance from this role is required to resolve the claim.

Section 4. Using the Process Designer

Extending the template using Process Designer

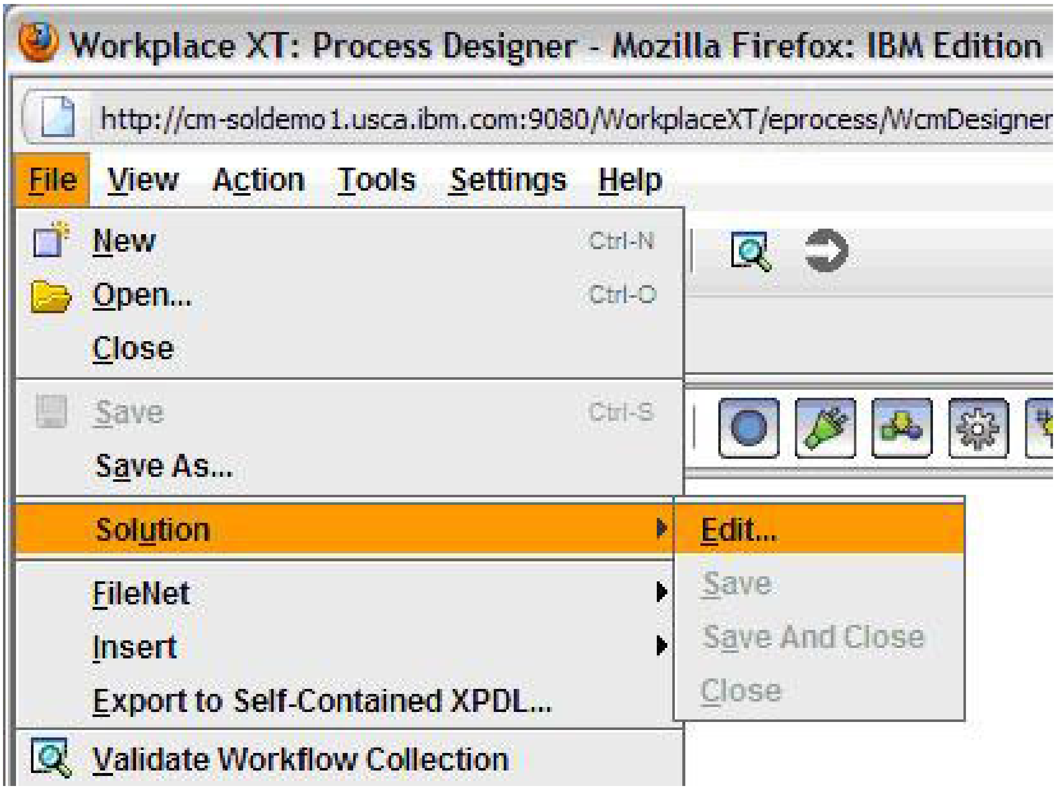
As the business analyst is implementing the solution, there will be instances where he will need to use Process Designer to implement additional required features. For this template, we have used Process Designer to integrate an eForm into the UI. Also, Process Designer is used to integrate ILOG JRules into the solution, which is discussed in more detail later in this tutorial.

Opening a solution in Process Designer

It is common to use functions or system fields to set values for case properties. To accomplish this, you must step outside of the Case Builder to work in Process Designer. Fortunately, there is good integration between these tools to make this transition seamless.

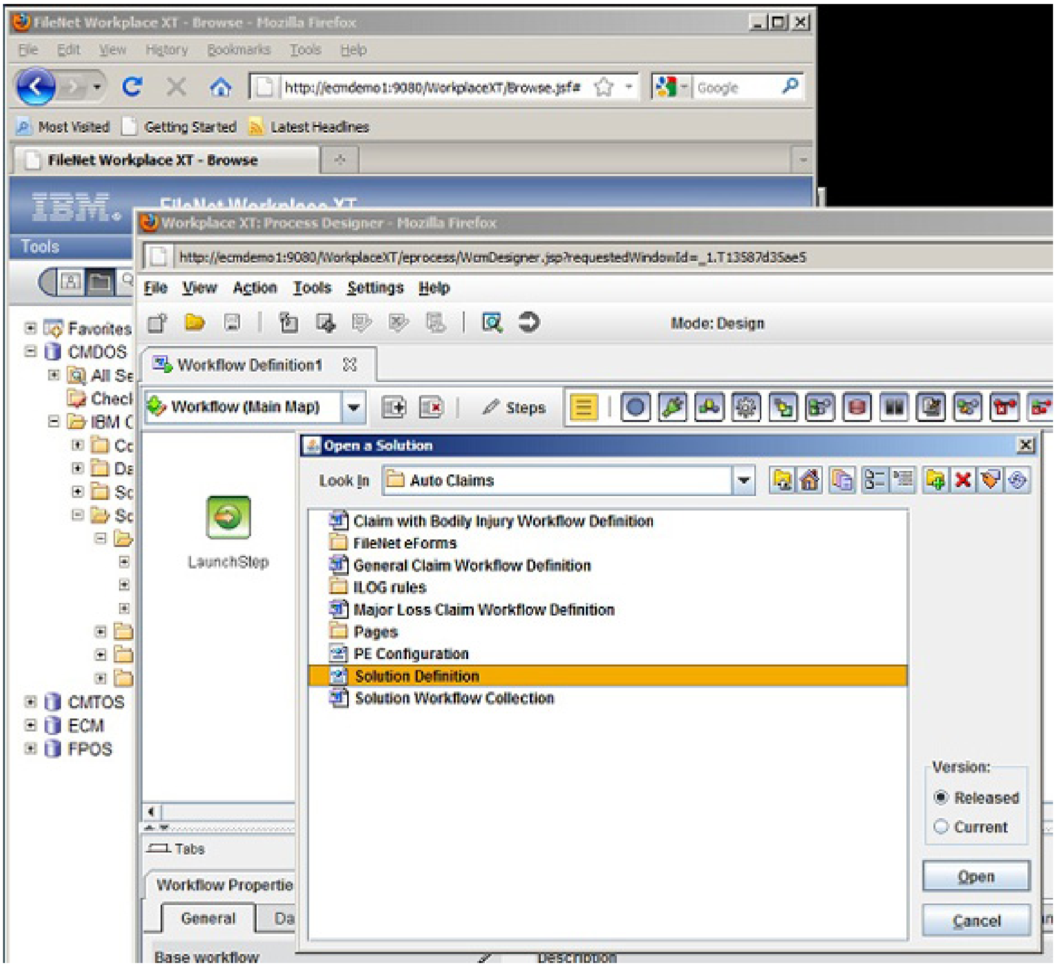
There are a few things you should be aware of as you move between Case Builder and Process Designer:

- Before opening a solution in Process Designer, make sure you have saved and closed the solution in Case Builder. You are likely to overwrite changes if both tools have the solution open. Make sure your solution doesn't appear as "checked out" before opening it in Process Designer.
- When opening the solution in Process Designer, make sure you use the **File > Solution > Edit** menu option.
  Figure 12. Editing a Case Manager Solution in Process Designer


Workplace XT: Process Designer - Mozilla Firefox: IBM Edition
http://cm-soldemo1.usca.ibm.com:9080/WorkplaceXT/eprocess/WcmDesigner
File
View
Action
Tools
Settings
Help
New Ctrl-N
Open... Ctrl-O
Close
Save Ctrl-S
Save As...
Solution
Edit...
Save
Save And Close
Close
FileNet
Insert
Export to Self-Contained XPDL...
Validate Workflow Collection

Navigate to the **Solution** directory, and select the **Solution Definition** file (shown in Figure 13) so the entire context of the process aspects of the solution is provided to Process Designer.

Figure 13. Editing a Case Manager Solution in Process Designer


FileNet Workplace XT - Browse - Mozilla Firefox
http://ecmdemo1:9080/WorkplaceXT/Browse.jsf#
Workplace XT: Process Designer - Mozilla Firefox
http://ecmdemo1:9080/WorkplaceXT/eprocess/WcmDesigner.jsp?requestedWindowId=_1.T13587d35ae5
File View Action Tools Settings Help
Mode: Design
Workflow Definition1
Workflow (Main Map)
Steps
LaunchStep
Favorites
CMDOS
CMTOS
ECM
FPOS
Tabs
Workflow Properties
General
Base workflow
Description

Open a Solution

Look In: Auto Claims

- Claim with Bodily Injury Workflow Definition
- FileNet eForms
- General Claim Workflow Definition
- ILOG rules
- Major Loss Claim Workflow Definition
- Pages
- PE Configuration
- Solution Definition
- Solution Workflow Collection

Version: ● Released ○ Current

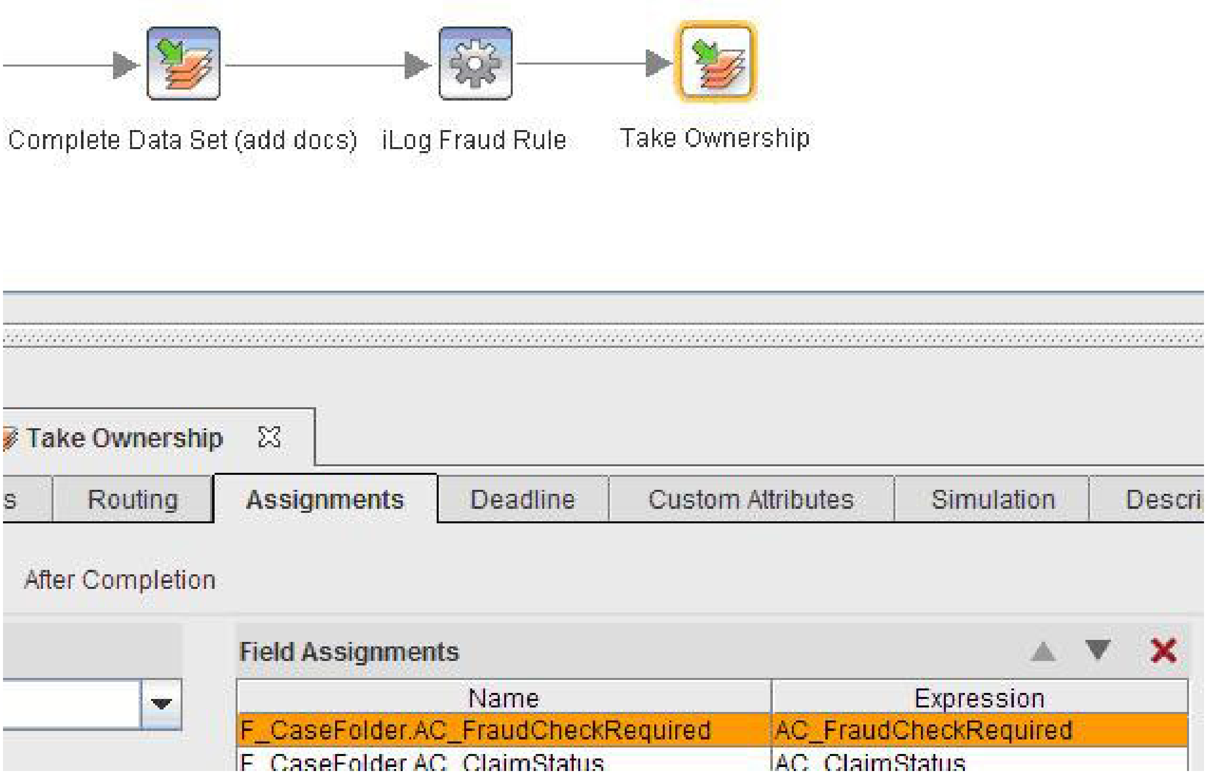
Open Cancel

- Case properties are referenced by their symbolic names in Process Designer. This is actually true anywhere outside of Case Builder. As an example, if your Auto Claims solution has a solution identifier of "AC," then all properties, document types, roles, etc. will have a prefix of "AC_." The symbolic name of a PolicyNumber property defined in Case Builder would be AC_PolicyNumber (note the case-sensitivity and removal of spaces).
- Case properties, case instance properties, and workflow data fields are different entities. As described earlier, when a case type is defined in Case Builder, it is modeled as a case folder in the FileNet P8 Content Engine. When you define a property within a case type, it is modeled as a property of the case folder. This is where property values for a case instance are persisted. Workflow data fields must be initialized with the case data properties. Case Builder does this for you automatically, but when you are using Process Designer to create steps, you must do this manually.

Let's look at an example.

Figure 14 shows the last three steps of the *record claim* task after importing the solution into Process Designer. It shows the step where the CSR completes the set of data for the execution of the claim case, a system step in which ILOG is invoked as a web service, and the step where the claims adjuster takes ownership of the claim. The field assignment is executed before the claims adjuster executes the last step.

Figure 14. Copying a property value to a case property



Name	Expression
F_CaseFolder.AC_FraudCheckRequired	AC_FraudCheckRequired
F_CaseFolder.AC_ClaimStatus	AC_ClaimStatus

Our goal is to instantiate the *evaluate potential fraud* task, based on the outcome of a rule executed by ILOG. We created the *evaluate potential fraud* task as an automatic task and gave it the following precondition:

```
AC_FraudCheckRequired = TRUE
```

When working in the Process Designer, the case property AC_FraudCheckRequired is available as the following, since case properties persist as properties of a case folder:

```
F_CaseFolder.AC_FraudCheckRequired
```

So we need to set the value of the case property

F_CaseFolder.AC_FraudCheckRequired by using the following assignment:

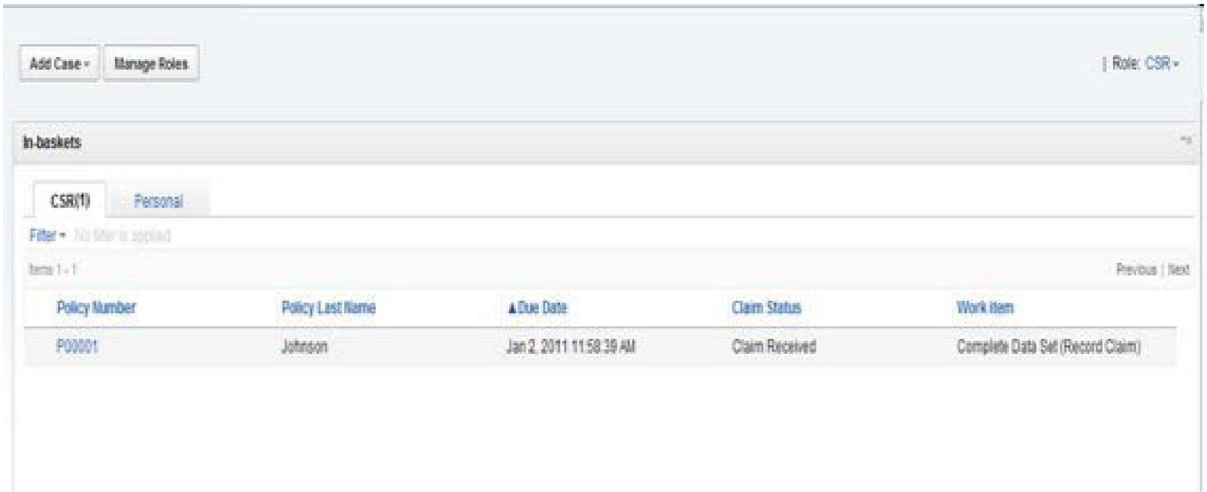
```
F_CaseFolder.AC_FraudCheckRequired= AC_FraudCheckRequired
```

(Refer to Figure 14.)

Extending Auto Claims in Process Designer setting in-basket properties

Within Auto Claims, the customer service in-basket looks like Figure 15.

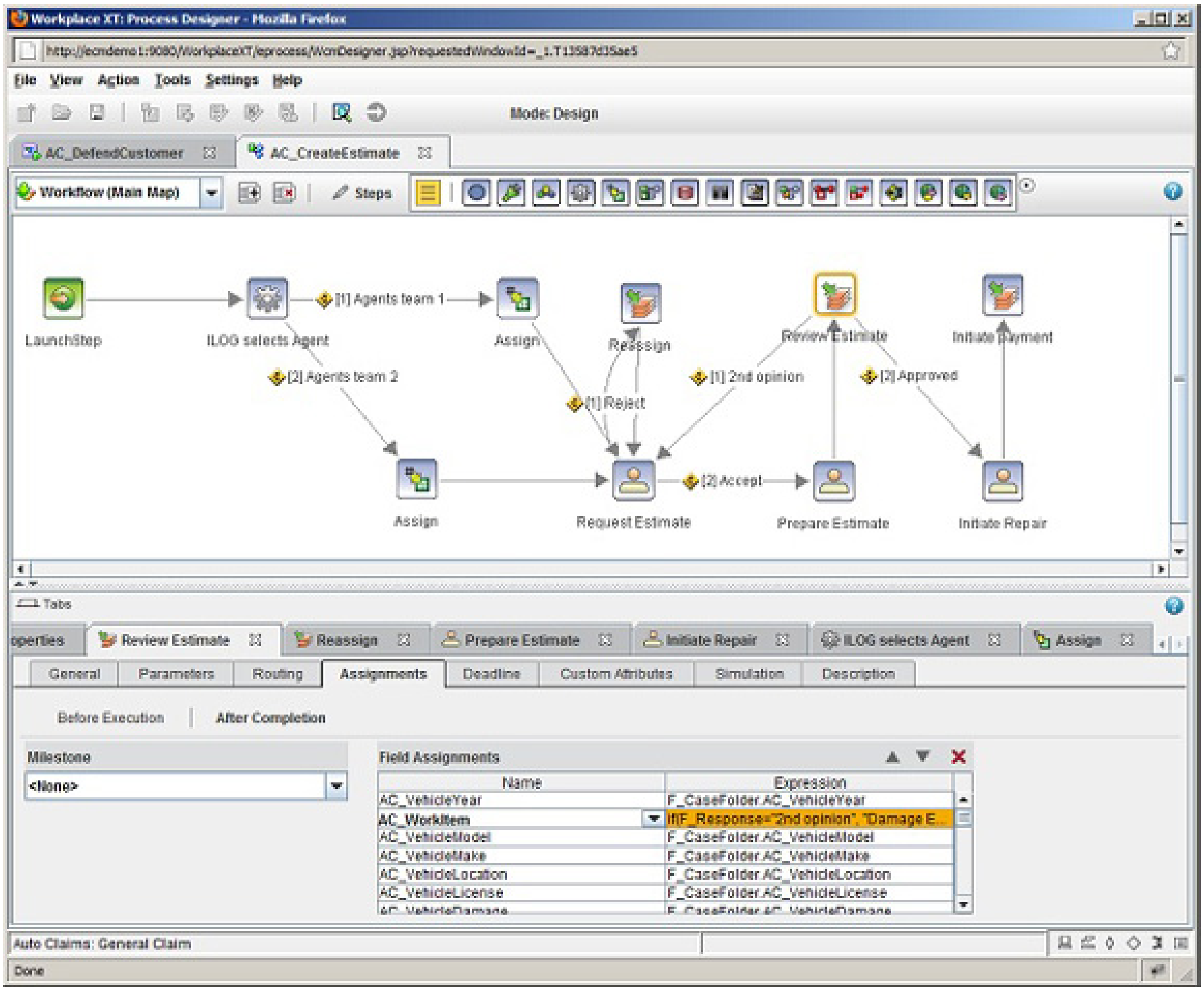
Figure 15. The columns of the CSR in-basket



Policy Number	Policy Last Name	▲Due Date	Claim Status	Work item
P00001	Johnson	Jan 2, 2011 11:58:39 AM	Claim Received	Complete Data Set (Record Claim)

The values for column Work Item is populated by manipulating case properties within Process Designer on a step-by-step basis. To see how this is done, consider the *create estimate* task flow.

Figure 16. Assigning property values in Process Designer



Milestone

<None>

Field Assignments

Name	Expression
AC_VehicleYear	F_CaseFolder.AC_VehicleYear
AC_WorkItem	If(F_Response="2nd opinion", "Damage E...
AC_VehicleModel	F_CaseFolder.AC_VehicleModel
AC_VehicleMake	F_CaseFolder.AC_VehicleMake
AC_VehicleLocation	F_CaseFolder.AC_VehicleLocation
AC_VehicleLicense	F_CaseFolder.AC_VehicleLicense
AC_VehicleDamage	F_CaseFolder.AC_VehicleDamage

The *review estimate* step is selected, and we are looking at the assignments that occur after the step is completed. The work item property is highlighted (note the symbolic name is used: AC_WorkItem). In this example, we are updating the value of the work item property, based on the case worker's response when completing the step *review estimate*:

```
If(F_Response="2nd opinion", "Damage Estimation Request","Initiate Repair")
```

Section 5. ILOG JRules system

Integrating the ILOG JRules system

Two simple ILOG JRules rules have been implemented for demonstrative purposes:

- **Instantiate a task, based on the outcome of a rule**
  After the CSR has recorded the data for the claim, an ILOG rule will start an *evaluate potential fraud* task, based on the data given by the customer and data imported from the policy database.

  The Auto Claim sample solution implements an example scenario where the company wants to execute a standard fraud investigation task when a young person files a claim for a new car of a certain brand. The rule for this scenario is shown in Listing 1.

  Listing 1. The business rule that instantiates the evaluate potential fraud task

```
definitions
    set 'Lower Limit Estimate' to 4999;
    set 'Upper Limit Policy Start Date' to 01/01/2008;
if
    all of the following conditions are true:
        - 'Estimate' is more than 'Lower Limit Estimate'
        - 'Policy Start Date' is after 'Upper Limit Policy Start Date',
then
    print "Evaluate for fraud";
    set 'Is Fraud Check Required' to true;
else
    print "Do not evaluate";
    set 'Is Fraud Check Required' to false;
```

  After the CSR has collected all data and has submitted the task to the claim adjuster, in the background, this rule has been executed to determine if this type of claim should undergo a standard fraud check. If so, then ILOG will launch the *evaluate potential fraud* task, and the claim adjuster may decide to put the *record claim* task on hold, awaiting the outcome of the fraud check.

- **Rule-based case assignment**
  ILOG JRules is used to help the CSR select an agent who can estimate the cost to repair the damage. In real life, this will be somebody located near the site of the accident, has knowledge of the type of vehicle that needs to be inspected, and is available to take on this job (based on the number of items in the agent's queue).

  The Auto Claim sample assumes that there are two agents. One of them inspects cars of Japanese make, while the other inspects all other cars. The rule for this scenario is shown in Listing 2.

  Listing 2. The business rule used to assign a task to an agent

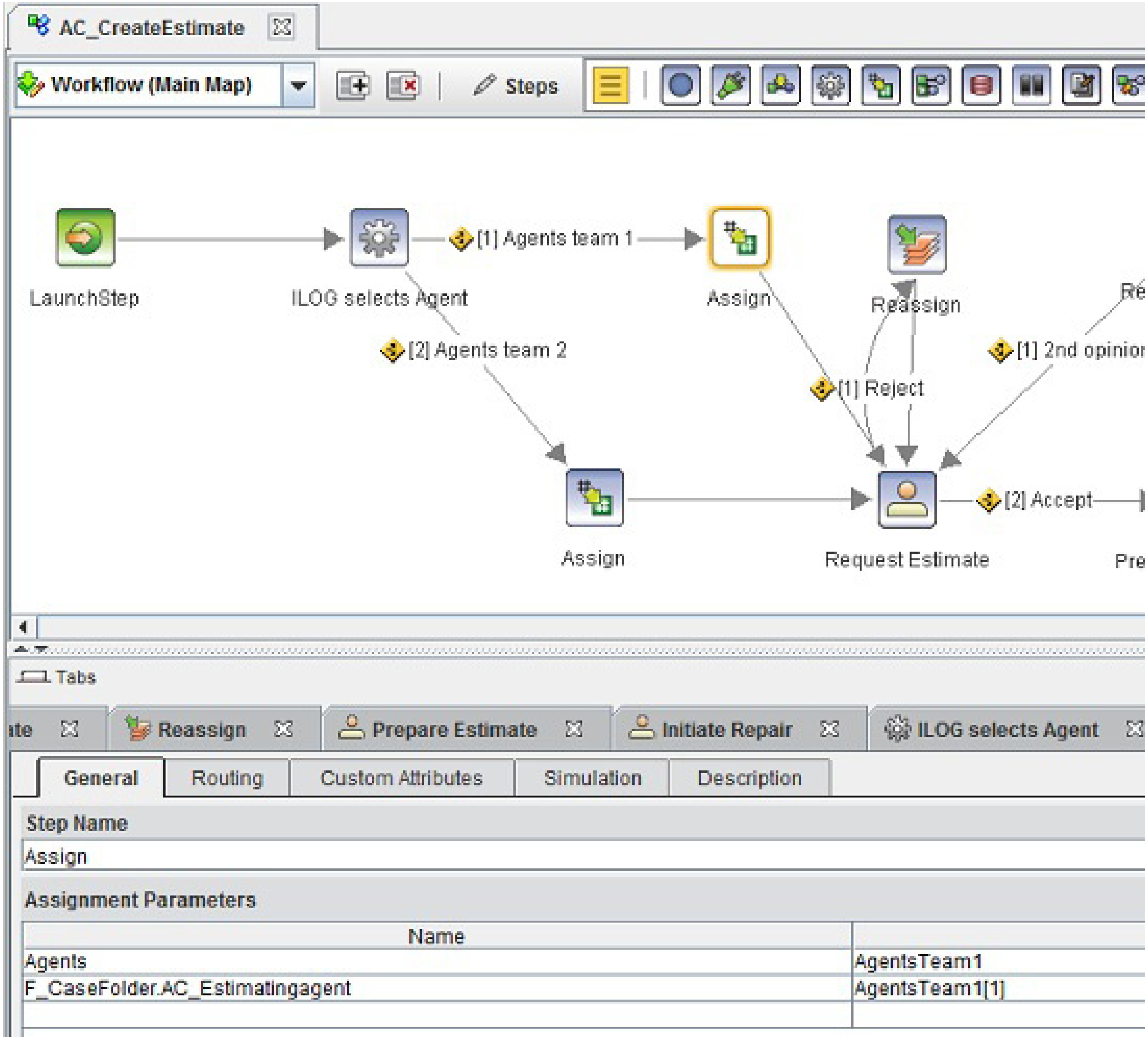
```
definitions
    set 'ACM Auto List 1' to {"Honda","Toyota","Mazda","Acura","Lexus"}'
if
    'Vehicle Make' is one of 'ACM Auto List 1'
then
    print "Japanese car";
    set 'Agent for Estimate' to "sroth';
else
    print "not a Japanese car";
    set 'Agent for Estimate' to "sfranks";
```

With the integration into ACM, `'Agent for Estimate'` will add one user to the workgroup Agents. This user will then execute the *estimate* task.

When building a case management solution, the integration of ILOG JRules is achieved the same way it is done using BPM: by using ILOG as a web service. In addition, we use assignments to pass the values provided by ILOG back to the case properties. As discussed in a previous section, in case of the ILOG fraud rule, ILOG fills the AC_FraudCheckRequired property, which we pass back to the case property F_CaseFolder.AC_FraudCheckRequired.

The case property F_CaseFolder.AC_FraudCheckRequired is a precondition for the *evaluate potential fraud* task. Therefore, that task will start if ILOG sets the case property to TRUE.

Figure 17. Adding the agent to the Agents workgroup



Step Name

Assign

Assignment Parameters

Name	
Agents	AgentsTeam1
F_CaseFolder.AC_Estimatingagent	AgentsTeam1[1]

In the *create estimate* task, we use ILOG to select an agent to prepare the estimate, based on the make of the vehicle. ILOG fills a workflow parameter agent in the system step *ILOG selects agent*, which we then use to determine which user(s) will be in the Agents workgroup, by copying the users from a pre-defined group. We could also have used the `Agents[1]=Agent` assignment to assign the user to the Agents workgroup. This requires that the ILOG rule output is the correct user name, instead of a variable that routes the work.

We then allow the claims adjuster to reassign the work to a different agent in the *Reassign* step in case the agent cannot do the damage estimation after all.

The UI assets

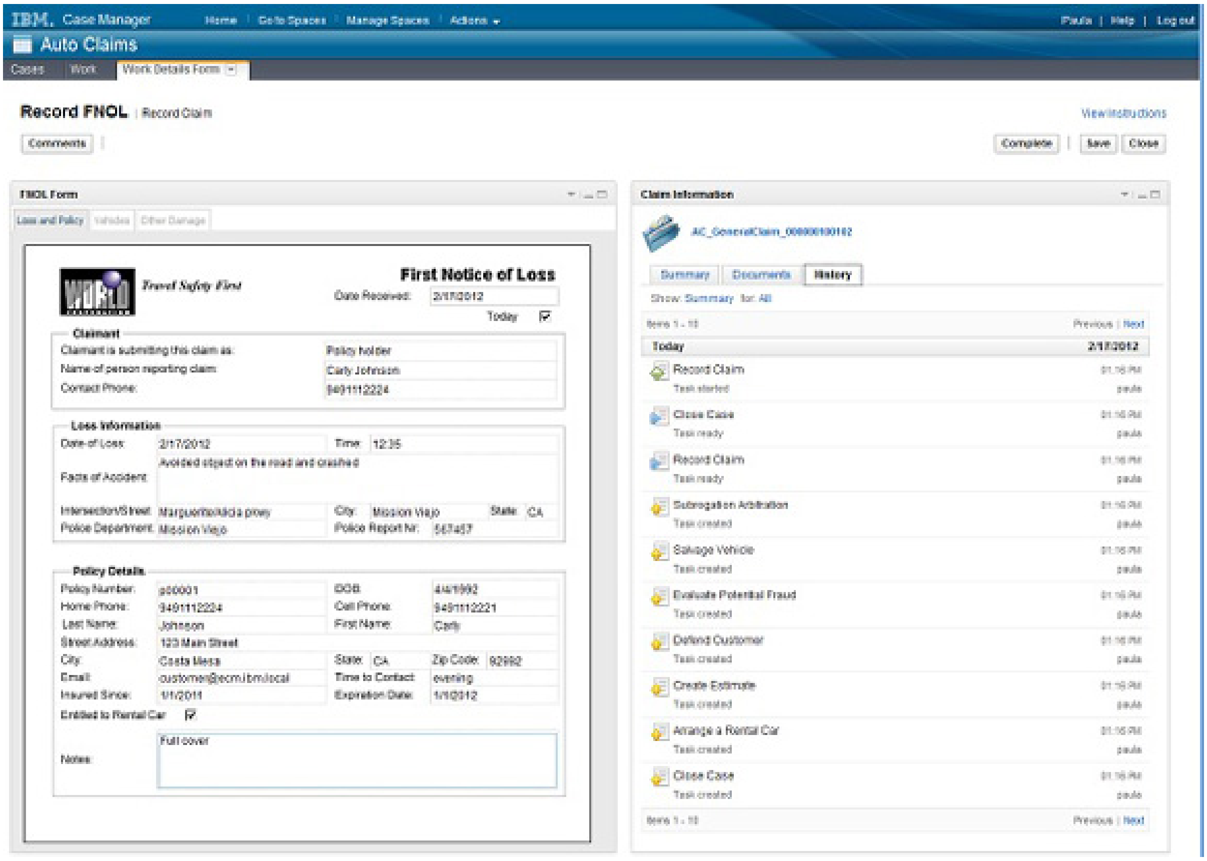
Customizing the Work Details page using an eForm

In release 5.1, IBM Case Manager supports IBM Forms, as well as FileNet eForms. In the Auto Claims template we use FileNet eForms.

Within the Auto Claims template, eForms are used to provide case workers with a consolidated view of the properties of the case. It provides flexibility in building a UI consistent with the company's branding and is most convenient for the case worker.

eForms can also be used to provide a stateful view of the properties of the case as the case moves through different steps and among workers. Figure 18 shows the user interface for the customer service representative as he gathers the initial details about the case.

Figure 18. The UI of the Auto Claims sample solution



First Notice of Loss

Date Received:	2/17/2012
Today	☑

Claimant

Claimant is submitting this claim as:	Policy holder
Name of person reporting claim:	Carly Johnson
Contact Phone:	9491112224

Loss Information

Date of Loss:	2/17/2012	Time:	12:35		
Facts of Accident:	Avoided object on the road and crashed				
Intersection/Street:	Marguerite/Alicia pkwy	City:	Mission Viejo	State:	CA
Police Department:	Mission Viejo	Police Report Nr:	567457		

Policy Details

Policy Number:	p00001	DOB:	4/4/1992		
Home Phone:	9491112224	Cell Phone:	9491112221		
Last Name:	Johnson	First Name:	Carly		
Street Address:	123 Main Street				
City:	Costa Mesa	State:	CA	Zip Code:	92992
Email:	customer@ecm.ibm.local	Time to Contact:	evening		
Insured Since:	1/1/2011	Expiration Date:	1/1/2012		
Entitled to Rental Car:	☑				
Notes:	Full cover				

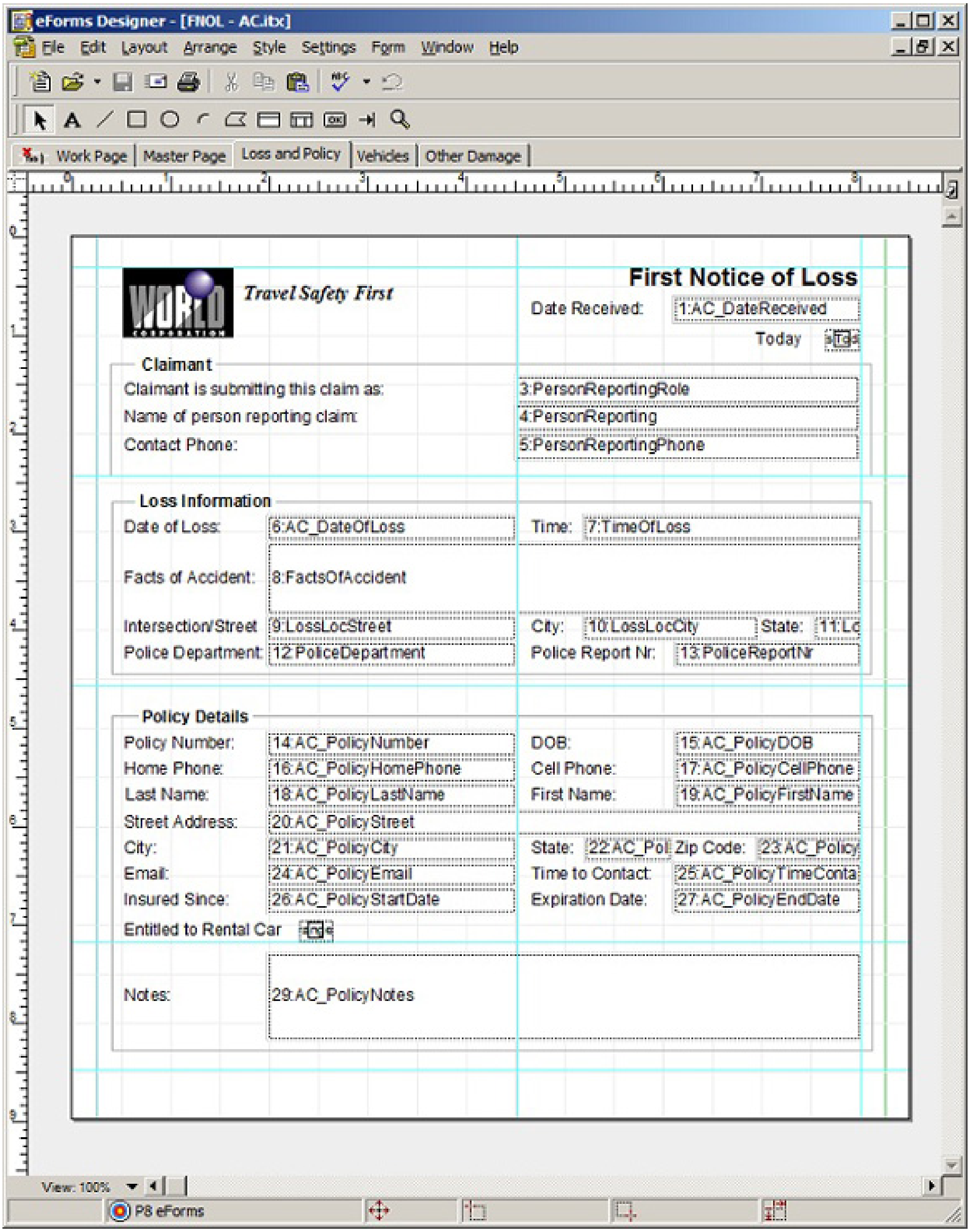
AC_GeneralClaim_000000100182

Today	2/17/2012
Record Claim Task started	01:16 PM paula
Close Case Task ready	01:16 PM paula
Record Claim Task ready	01:16 PM paula
Subrogation Arbitration Task created	01:16 PM paula
Salvage Vehicle Task created	01:16 PM paula
Evaluate Potential Fraud Task created	01:16 PM paula
Defend Customer Task created	01:16 PM paula
Create Estimate Task created	01:16 PM paula
Arrange a Rental Car Task created	01:16 PM paula
Close Case Task created	01:16 PM paula

This is a customized Work Details eForm page. The widget on the left is an eForm, which replaces the Case Data widget. The CSR can enter the policy number or the customer's phone number, and the eForm will then populate some fields of the customer information by querying a database. The CSR can continue to fill out more details of the claim.

Integrating FileNet eForms with IBM Case Manager is easy. eForm fields and case properties can simply be mapped using the symbolic name of the case property as the eForm field name. We only map a subset of the eForm fields with case Manager properties, as illustrated in Figure 19. The eForm policies that are used in P8 BPM are not used in IBM Case Manager.

Figure 19. The FileNet eForms Designer application


eForms Designer - [FNOL - AC.itx]
File Edit Layout Arrange Style Settings Form Window Help
Work Page | Master Page | Loss and Policy | Vehicles | Other Damage
View: 100%
P8 eForms

Travel Safety First

First Notice of Loss

Date Received: 1:AC_DateReceived

Today

Claimant

Claimant is submitting this claim as:	3:PersonReportingRole
Name of person reporting claim:	4:PersonReporting
Contact Phone:	5:PersonReportingPhone

Loss Information

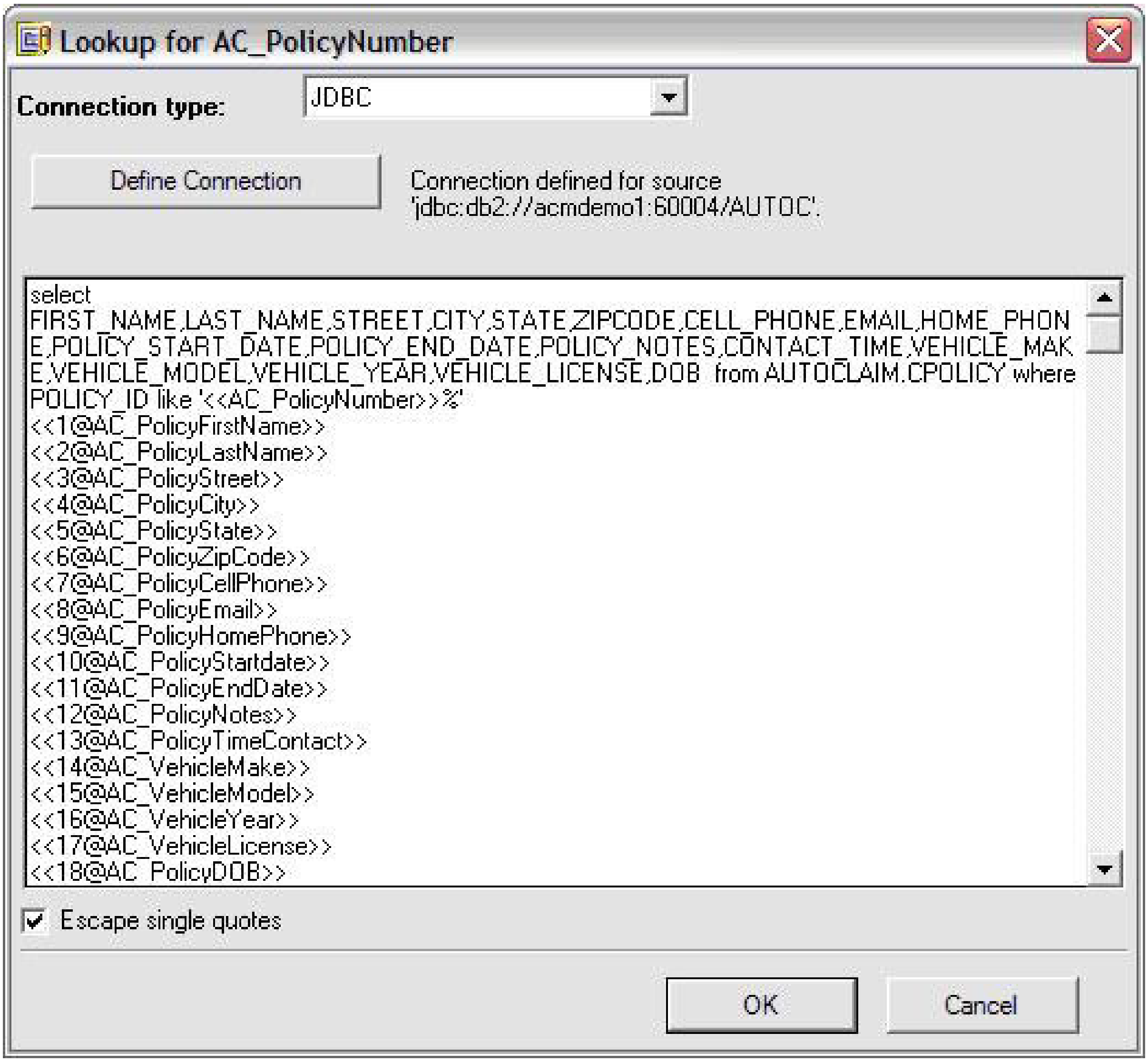
Date of Loss:	6:AC_DateOfLoss	Time:	7:TimeOfLoss		
Facts of Accident:	8:FactsOfAccident				
Intersection/Street	9:LossLocStreet	City:	10:LossLocCity	State:	11:Lo
Police Department:	12:PoliceDepartment	Police Report Nr:	13:PoliceReportNr		

Policy Details

Policy Number:	14:AC_PolicyNumber	DOB:	15:AC_PolicyDOB		
Home Phone:	16:AC_PolicyHomePhone	Cell Phone:	17:AC_PolicyCellPhone		
Last Name:	18:AC_PolicyLastName	First Name:	19:AC_PolicyFirstName		
Street Address:	20:AC_PolicyStreet				
City:	21:AC_PolicyCity	State:	22:AC_Pol	Zip Code:	23:AC_Policy
Email:	24:AC_PolicyEmail	Time to Contact:	25:AC_PolicyTimeConta		
Insured Since:	26:AC_PolicyStartDate	Expiration Date:	27:AC_PolicyEndDate		
Entitled to Rental Car					
Notes:	29:AC_PolicyNotes				

The First Notice of Loss for the auto claim is recorded in a FNOL eForm. This eForm loads the policy data from a policy information database, including vehicle details, as the CSR types the policy number, the home phone number of the policy holder, or the last name of the policy holder.

Figure 20. Database lookups in the FileNet eForms Designer application



Connection type: JDBC

Connection defined for source 'jdbc:db2://acmdemo1:60004/AUTOC'.

```
select
FIRST_NAME,LAST_NAME,STREET,CITY,STATE,ZIPCODE,CELL_PHONE,EMAIL,HOME_PHON
E,POLICY_START_DATE,POLICY_END_DATE,POLICY_NOTES,CONTACT_TIME,VEHICLE_MAK
E,VEHICLE_MODEL,VEHICLE_YEAR,VEHICLE_LICENSE,DOB  from AUTOCLAIM.CPOLICY where
POLICY_ID like '<<AC_PolicyNumber>>%'
<<1@AC_PolicyFirstName>>
<<2@AC_PolicyLastName>>
<<3@AC_PolicyStreet>>
<<4@AC_PolicyCity>>
<<5@AC_PolicyState>>
<<6@AC_PolicyZipCode>>
<<7@AC_PolicyCellPhone>>
<<8@AC_PolicyEmail>>
<<9@AC_PolicyHomePhone>>
<<10@AC_PolicyStartdate>>
<<11@AC_PolicyEndDate>>
<<12@AC_PolicyNotes>>
<<13@AC_PolicyTimeContact>>
<<14@AC_VehicleMake>>
<<15@AC_VehicleModel>>
<<16@AC_VehicleYear>>
<<17@AC_VehicleLicense>>
<<18@AC_PolicyDOB>>
```

☑ Escape single quotes

This is done using the lookup capability in eForms, using a JDBC connection. This capability is not new and can be used in a BPM application as well.

Retrieving the data from the policy information database has multiple purposes:

- It allows the CSR to verify that the policy is valid and that there are no issues.
- It limits the amount of data that needs to be typed in the FNOL form.
- It automates starting a potential new task, based on information in the policy information database. If the policy includes entitlement to a rental

car, then the *arrange a rental car* task will automatically be started after the CSR completes the form.

- It allows you to review information in the policy database even if it is not relevant for the processing of the claim and doesn't need to propagate to case properties.

FileNet eForms also provides capabilities to manipulate field data, which is demonstrated by selecting **Policy holder** for the role of the claimant or the driver. This selection will copy the policy holder name and phone number into the appropriate eForm fields, while allowing the CSR to make further changes to those fields. As an example, the field 2:PersonReporting uses the following calculation for its value:

```
    If PersonReportingRole = "Policy holder" Then
      Concat (AC_PolicyFirstName," ",AC_PolicyLastName)
  End
```

As the FNOL eForm is completed, several fields of the form will populate case properties, making the information available throughout the claim process. Some of the case properties are used as a precondition for starting tasks, and the template will demonstrate that. The case data will also populate some fields in the other forms used in the solution.

A FNOL form data document will be stored in the case folder, making it available to all roles involved in the claim process.

An agent (assessor) will then be selected to estimate the damage. The Request for Estimate eForm used to request the estimate is pre-filled with the case data that was recorded. The agent will only receive the information required to do his job. The agent can accept or reject the request, based on his workload and expertise.

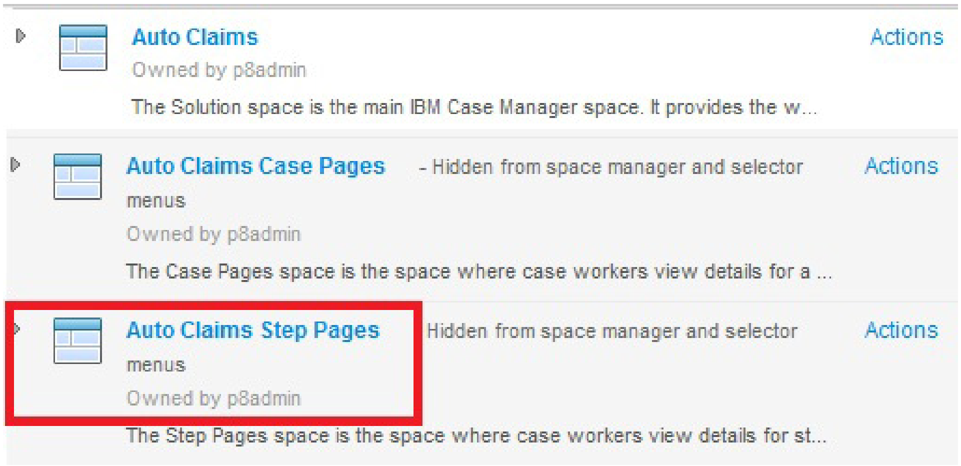
After the agent finishes and submits a Damage Estimate eForm, the claim adjuster will approve the estimate and ask the agent to initiate the repair process. If, however, the damage exceeds the day value of the vehicle, then a *salvage* task will be started.

Enabling eForms in IBM Case Manager

To use an eForm in the Auto Claims sample template, perform the following steps:

1. Add an eForm widget on the page where you want to use the eForm:

   a. In Case Manager Client, select **Space Manager** and open the **Step Pages Space** for your solution:
   Figure 21. Editing the Auto Claims solution pages



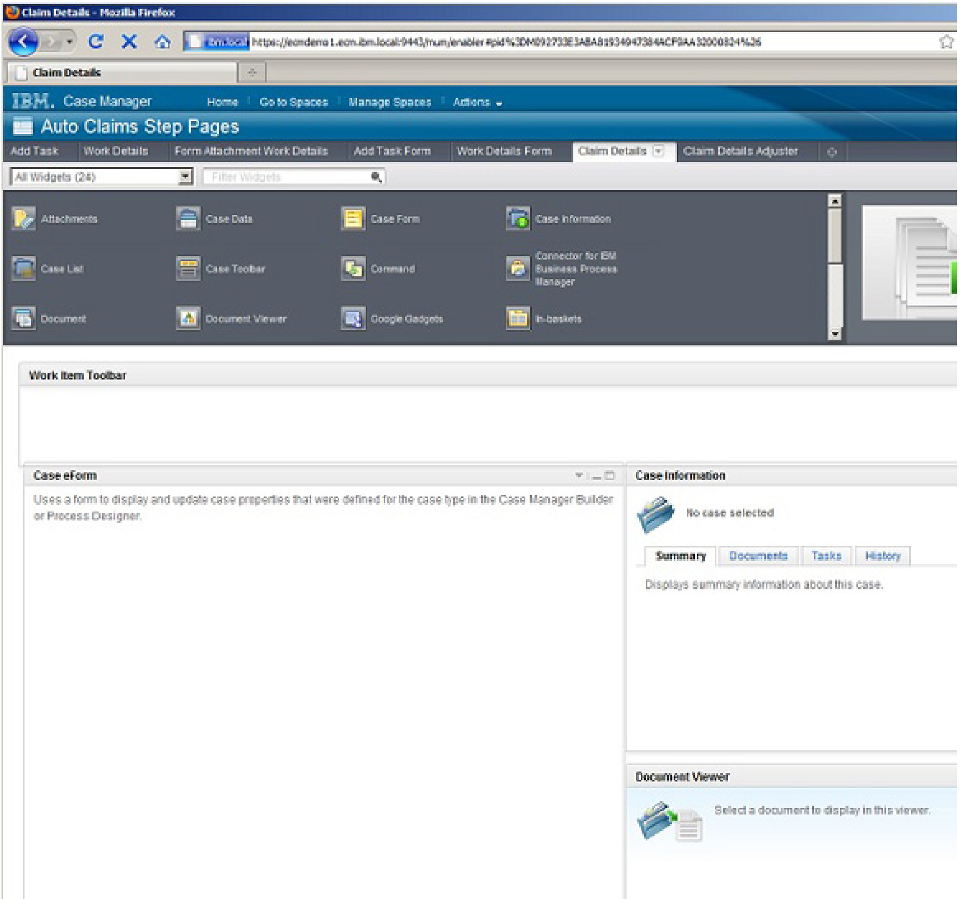
Auto Claims
Owned by p8admin
The Solution space is the main IBM Case Manager space. It provides the w...

Auto Claims Case Pages - Hidden from space manager and selector menus
Owned by p8admin
The Case Pages space is the space where case workers view details for a ...

Auto Claims Step Pages Hidden from space manager and selector menus
Owned by p8admin
The Step Pages space is the space where case workers view details for st...

b. Select the page where you want to add your eForm or create your new page.

Figure 22. Adding the eForm Widget to an Auto Claims page



Uses a form to display and update case properties that were defined for the case type in the Case Manager Builder or Process Designer.

Displays summary information about this case.

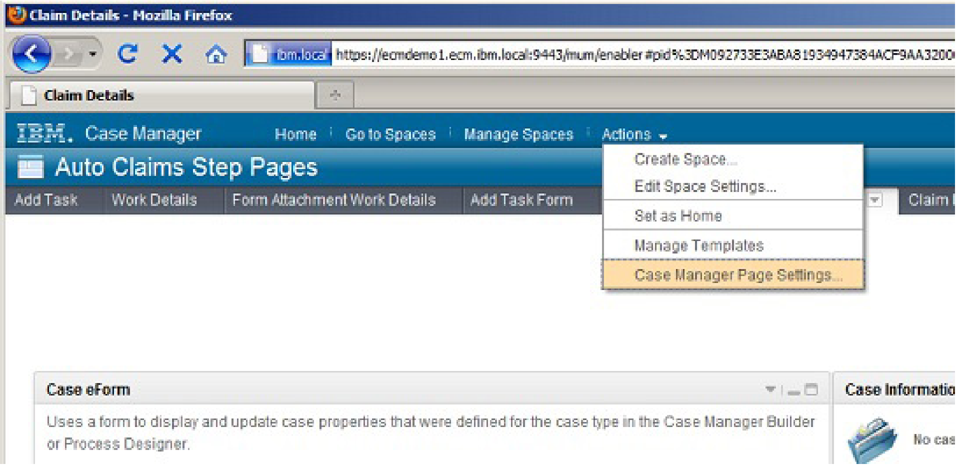
Select a document to display in this viewer.

For our Auto Claims sample template, we have designed a page with an eForm widget, a Case Information widget, and a Document Viewer widget.

Register the page so you can use the new or updated page as a step processor.

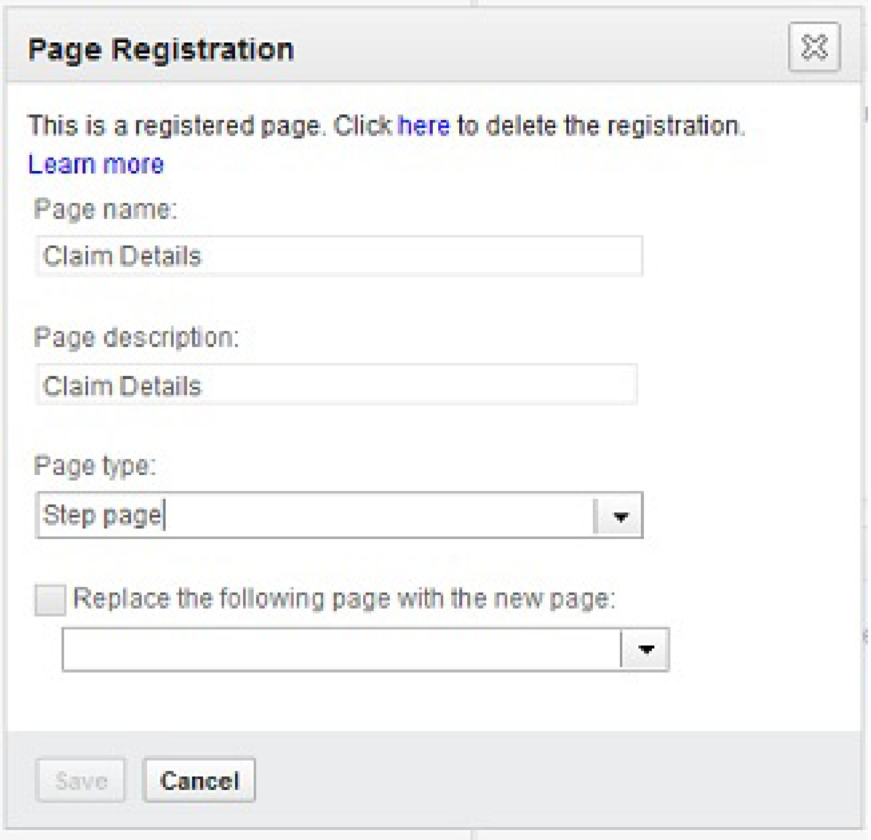
c. Still in Case Manager Client, select **Page Actions > Case Manager Page Settings**.

Figure 23. Registering the page


Claim Details - Mozilla Firefox
ibm.local
https://ecmdemo1.ecm.ibm.local:9443/mum/enabler#pid%3DM092733E3ABA81934947384ACF9AA3200
Claim Details
IBM. Case Manager
Home
Go to Spaces
Manage Spaces
Actions
Auto Claims Step Pages
Add Task
Work Details
Form Attachment Work Details
Add Task Form
Claim
Create Space...
Edit Space Settings...
Set as Home
Manage Templates
Case Manager Page Settings...
Case eForm
Uses a form to display and update case properties that were defined for the case type in the Case Manager Builder or Process Designer.
Case Informatio
No cas

d. Select **Step page** for Page Type, then click **Save** to register the page.

Figure 24. Registering the page as a step processor



Page Registration

This is a registered page. Click here to delete the registration.
Learn more

Page name:
Claim Details

Page description:
Claim Details

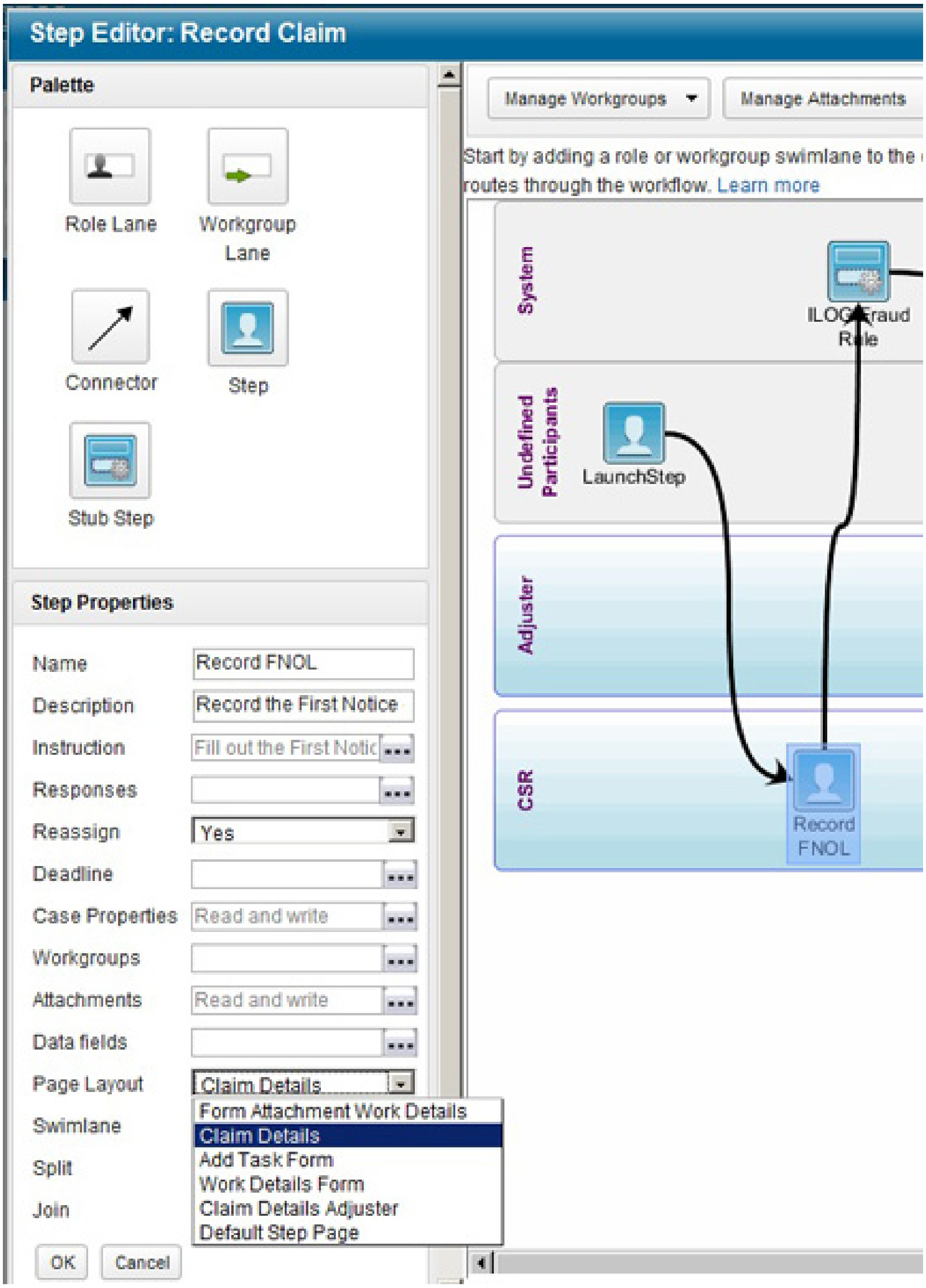
Page type:
Step page

□ Replace the following page with the new page:

2. Select the new page as your *step processor* for the step where the eForm is used:

   a. In IBM Case Manager Builder, open your solution and navigate to the task where you want to use your eForm.

   b. In Step Editor, select the step **Record FNOL** and update PageLayout with the page you just registered.
   Figure 25. Selecting a step processor in a task



Step Editor: Record Claim

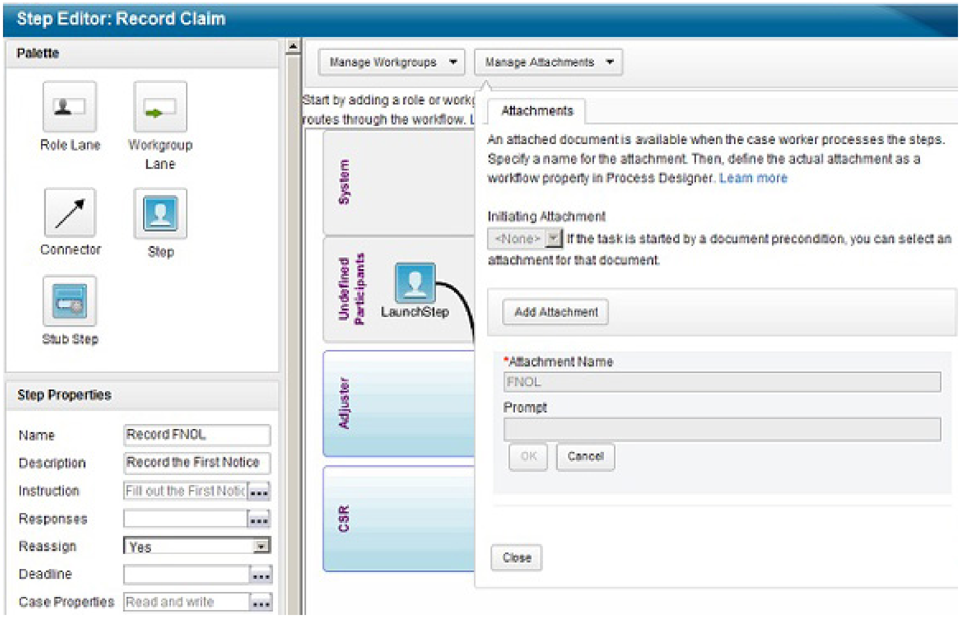
Start by adding a role or workgroup swimlane to the
routes through the workflow. Learn more

Step Properties

Name: Record FNOL
Description: Record the First Notice
Instruction: Fill out the First Notic
Reassign: Yes
Case Properties: Read and write
Attachments: Read and write
Page Layout: Claim Details

3. Create an attachment field in the task where you use the eForm, using Step Editor:

   a. To define a new attachment field FNOL_eForm, go to **Manage Attachments** at the top of the screen and define the FNOL_eForm attachment.
   Figure 26. Creating an attachment parameter



Start by adding a role or work… routes through the workflow. L…

An attached document is available when the case worker processes the steps. Specify a name for the attachment. Then, define the actual attachment as a workflow property in Process Designer. Learn more

Initiating Attachment
<None> If the task is started by a document precondition, you can select an attachment for that document.

*Attachment Name: FNOL

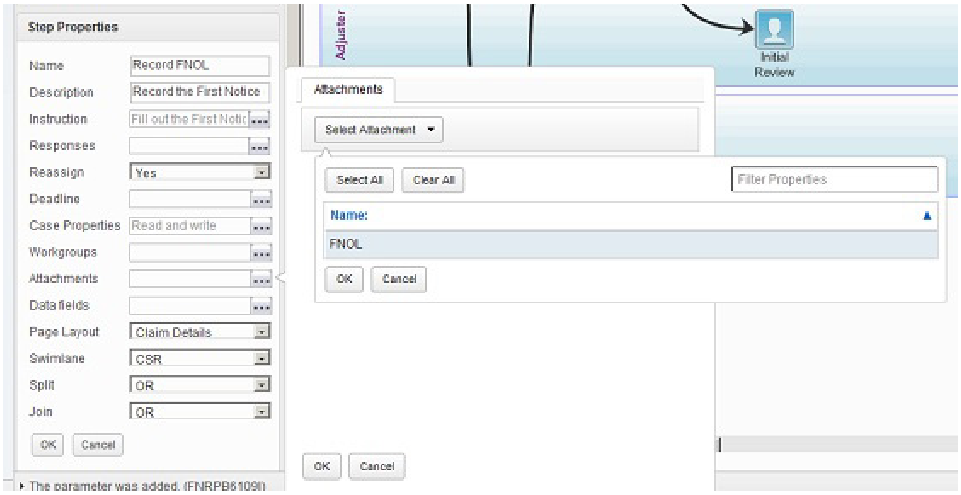
Prompt:

Name: Record FNOL
Description: Record the First Notice
Instruction: Fill out the First Notic
Responses:
Reassign: Yes
Deadline:
Case Properties: Read and write

4. Assign the new attachment field to the *record FNOL* step, where the eForm will be used:

   a. Select the step and the parameters, then navigate to the **Attachments** tab and select the attachment parameter you just created.

   Figure 27. Assigning an attachment parameter to a task



Name: Record FNOL
Description: Record the First Notice
Instruction: Fill out the First Notic
Responses:
Reassign: Yes
Deadline:
Case Properties: Read and write
Workgroups:
Attachments:
Data fields:
Page Layout: Claim Details
Swimlane: CSR
Split: OR
Join: OR

Name:
FNOL

The parameter was added. (FNRPB6109I)

5. Fill the attachment parameter with the eForm template that is going to be used.
   The IBM Case Manager Builder application does not allow users to pre-populate attachment fields in the first release. Attachment parameters are not case parameters; they only live for the duration of the task. In theory, you can assign an eForm in the application by displaying the

attachments widget in a previous step. However, we don't want to require the user to do this. The eForm needs to be readily available when the step is opened.

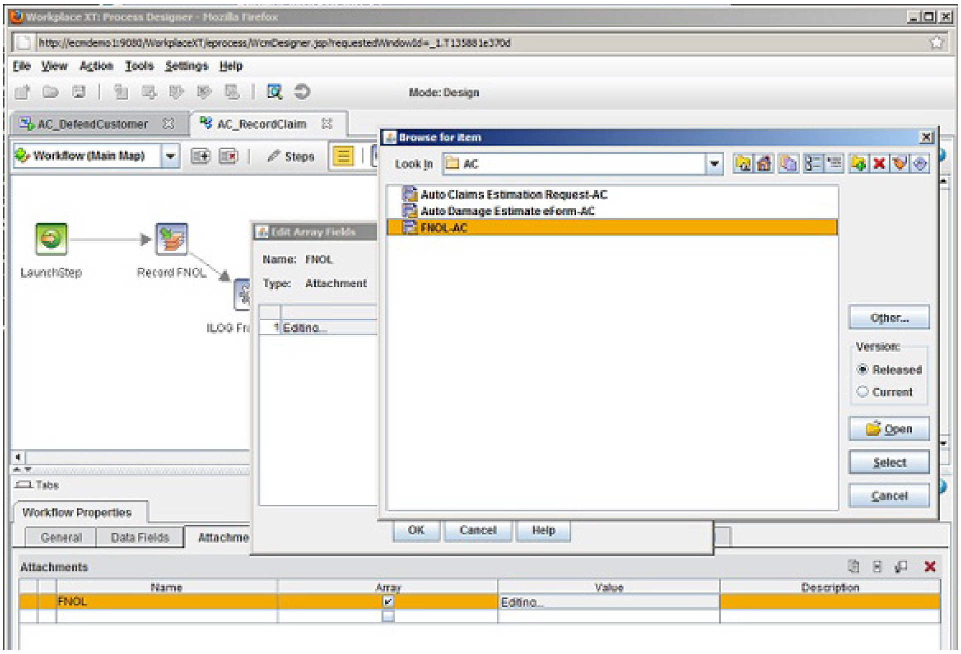
To solve this problem, we need to make a round-trip to Process Designer to assign the eForm to the attachment parameter:

a. Use the **File > Solution > Edit** menu option in Process Designer to open the task flows.

b. Select the **Workflow Properties** tab and then the **Attachments** tab.

c. Double-click on the value field of the multi-valued attachment field, browse the folder structure in the CMTOS object store, then select the eForm.

Figure 28. Assigning an eForm to an attachment parameter



Browse for Item, Look In: AC

- Auto Claims Estimation Request-AC
- Auto Damage Estimate eForm-AC
- FNOL-AC

Version: ● Released ○ Current

Edit Array Fields

Name: FNOL

Type: Attachment

1 Editing...

Attachments

Name	Array	Value	Description
FNOL	☑	Editing...	
	□		

Please note that it is a requirement to store the eForm in the object store where the application is deployed (CMTOS).

We performed the steps outlined above for the three eForms used in the Auto Claims sample template:

- The FNOL eForm used in the *record FNOL* step, in the *record claim* task
- The Request Estimate eForm used in the *request estimate* step, in the

create estimate task

- The Estimate eForm used in the *Prepare Estimate* step, in the *estimate* task

In the *create estimate* task, we used two attachment parameters containing the Request Estimate and Estimate eForms, and exposed each of them at the appropriate step.

Packaging and deployment

Case Manager object stores

IBM Case Manager requires a design object store to store the design artifacts and a target object store to store the runtime artifacts. By default, these object stores are named CMDOS and CMTOS.

The tools of the trade: CMAC and FileNet Deployment Manager

CMAC is an administration tool used to set up an ICM environment. It is also used to import solutions, create solutions from a template, or to promote a solution to be a template.

IBM FileNet Deployment Manager simplifies adding additional assets (like eForms) to the runtime solution. Specifically, this tool allows to redefine environment-specific data used by the application, such as user names and JDBC connections.

Section 6. Conclusion

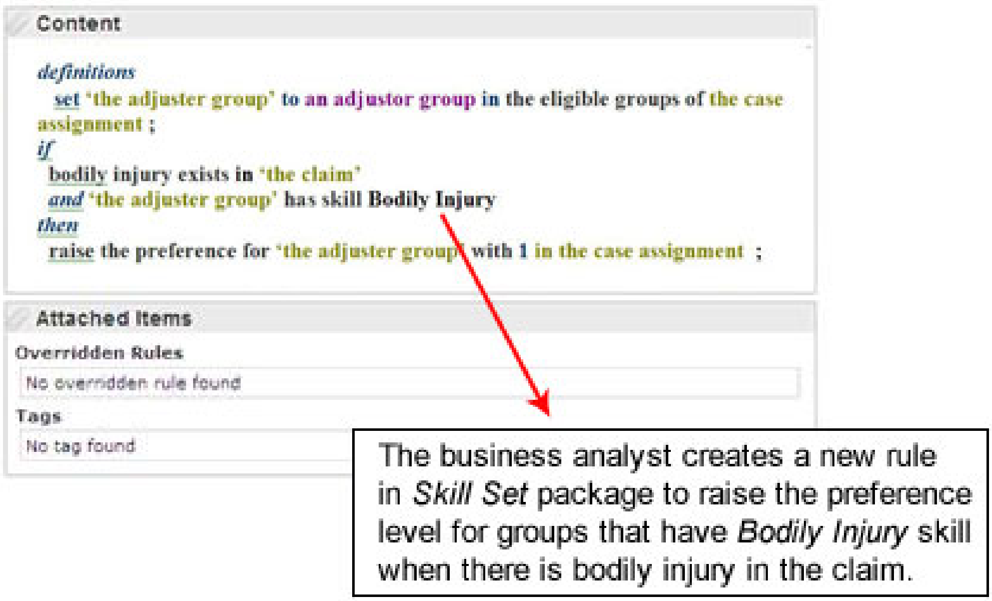
Solution templates are an important part of the IBM Case Manager platform. They provide an accelerator for customers and partners in building their own solutions using the platform. As part of the first release of IBM Case Manager, IBM has provided two sample templates to serve as learning tools and validation points for the platform. This tutorial has introduced the Auto Claims Management sample template, described the assets it contains, and provided insights on how particular assets within the template were built. After reading this tutorial, you should be more familiar with the notion of templates and have an understanding of how to use IBM Case Manager in conjunction with the other tools in the IBM FileNet P8 platform to build a complete case management solution.

Section 7. Addendum Advanced ILOG JRules assets

ILOG JRules has advanced capabilities that fit well with an Auto Claims industry solution and go far beyond what is illustrated in the Auto Claims industry solution template. The ILOG Services team can provide the following assets that can be used for a production solution that is based on the Auto Claims industry solution template:

- A *rule-based case assignment* asset that allows you to assign work based on various conditions, such as:
  - Regulatory guidelines
  - Insurance policy/contractual agreements
  - Severity of damage and injury
  - Adjuster availability and expertise
  - Workload balancing

Figure 29. Advanced task assignment using ILOG



Content

```
definitions
  set 'the adjuster group' to an adjustor group in the eligible groups of the case
assignment ;
if
  bodily injury exists in 'the claim'
  and 'the adjuster group' has skill Bodily Injury
then
  raise the preference for 'the adjuster group' with 1 in the case assignment ;
```

Attached Items

Overridden Rules

No overridden rule found

Tags

No tag found

The business analyst creates a new rule in *Skill Set* package to raise the preference level for groups that have *Bodily Injury* skill when there is bodily injury in the claim.

- A *dynamic forms* asset that provides the option of a self-service front end to your Auto Claim solution. Based on answers given, it guides the claimant through the process of filling out the First Notice of Loss form online before the CSR examines the claim data in the FNOL eForm provided with the Auto Claims industry solution template. This will provide the claim adjusters team with a more accurate data set, and it reduces the workload of the CSR team.

Downloads

Description	Name	Size	Download method
Artifacts of the Auto Claims sample template	AutoClaims.zip	12KB	HTTP

Information about download methods

Resources

Learn

- For more information, see IBM Case Manager Information Center to learn about IBM Case Manager and Take a guided tour of IBM Case Manager.
- Visit IBM FileNet P8 Information Center to learn about IBM FileNet P8.
- Learn more about IBM WebSphere ILOG JRules at the IBM WebSphere ILOG JRules Information Center.
- Visit the IBM Enterprise Content Management Application Center to get the resources you need to advance your skills with IBM Enterprise Content Management products.
- Learn more about Information Management at the developerWorks Information Management zone. Find technical documentation, how-to articles, education, downloads, product information, and more.
- Stay current with developerWorks technical events and webcasts.
- Follow developerWorks on Twitter.

Get products and technologies

- Build your next development project with IBM trial software, available for download directly from developerWorks.
- Now you can use DB2 for free. Download DB2 Express-C, a no-charge version of DB2 Express Edition for the community that offers the same core data features as DB2 Express Edition and provides a solid base to build and deploy applications.

Discuss

- Check out the developerWorks blogs and get involved in the developerWorks community.

About the author

Jos H. Olminkhof

Jos Olminkhof is a member of the product management team within the IBM Enterprise Content Management organization. His focus is on integrating the ECM portfolio into strategic solutions within various industries. He has been with IBM since 2006 and FileNet since 1996, is certified in design, administration and development with FileNet P8 products, and has worked within the Information Management space for

18 years.