

IBM Case Manager – Cases and Tasks

Business Partner Technical Enablement
IBM Enterprise Content Management Software





Course Overview

- The purpose of the presentation is to provide a detailed understanding of
 - IBM Case Manager Object Model
 - Case implementation in the Content Engine
 - Case Tasks
- We will also cover a the Split Case Feature

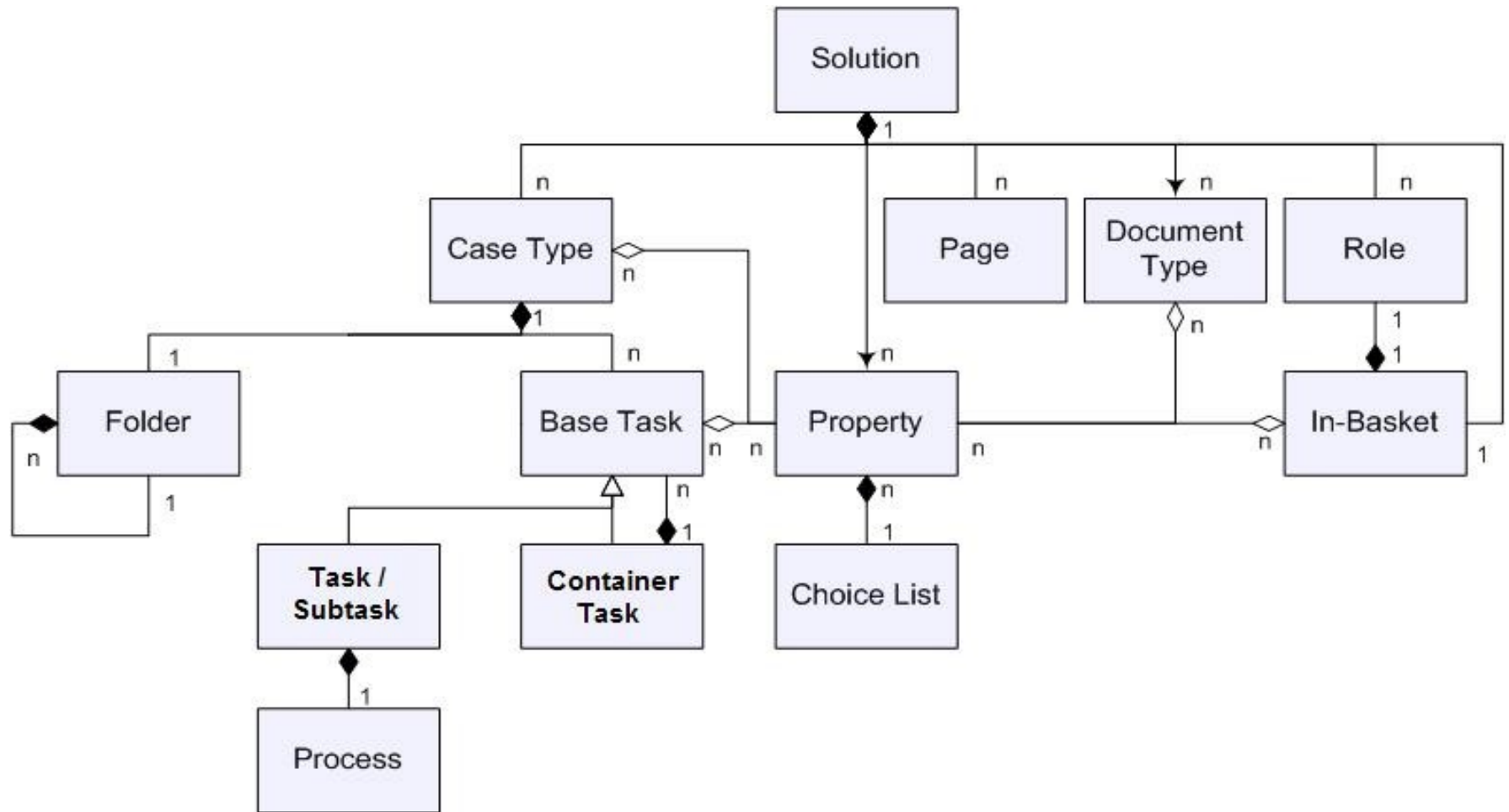
Topics

- Case Terminology
- Case Object Model Overview and Detailed Review
- Case structure and implementation in Content Engine
- Tasks
- Splitting Cases

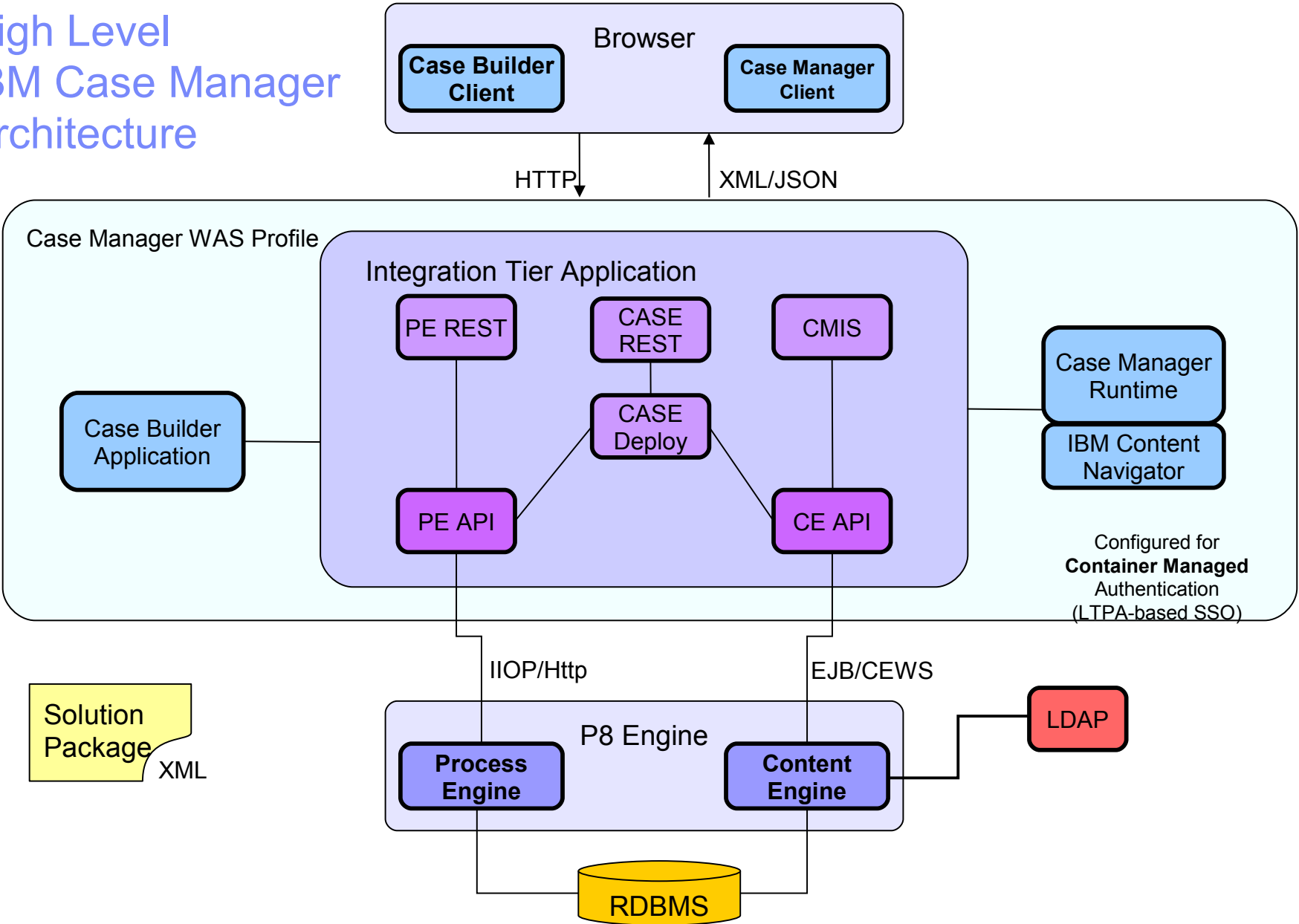
Glossary

- What is a Solution?
 - A set of user interfaces, content, and process definitions that provide the structure for the management of cases.
- What is a Case?
 - A set of related activities, content, processes, and collaboration artifacts used together to manage specific business activities.
- What is a Case Type?
 - A definition of a case
- What is a Task?
 - A list of items that need to be completed in the context of a case (i.e., a ToDo list).
- What is a Role?
 - A collection of user accounts associated with a specific business function. These are used to access a particular task or work queue step.
- What is a Document Type?
 - A document class in the P8 Content Engine.

ICM Object Model



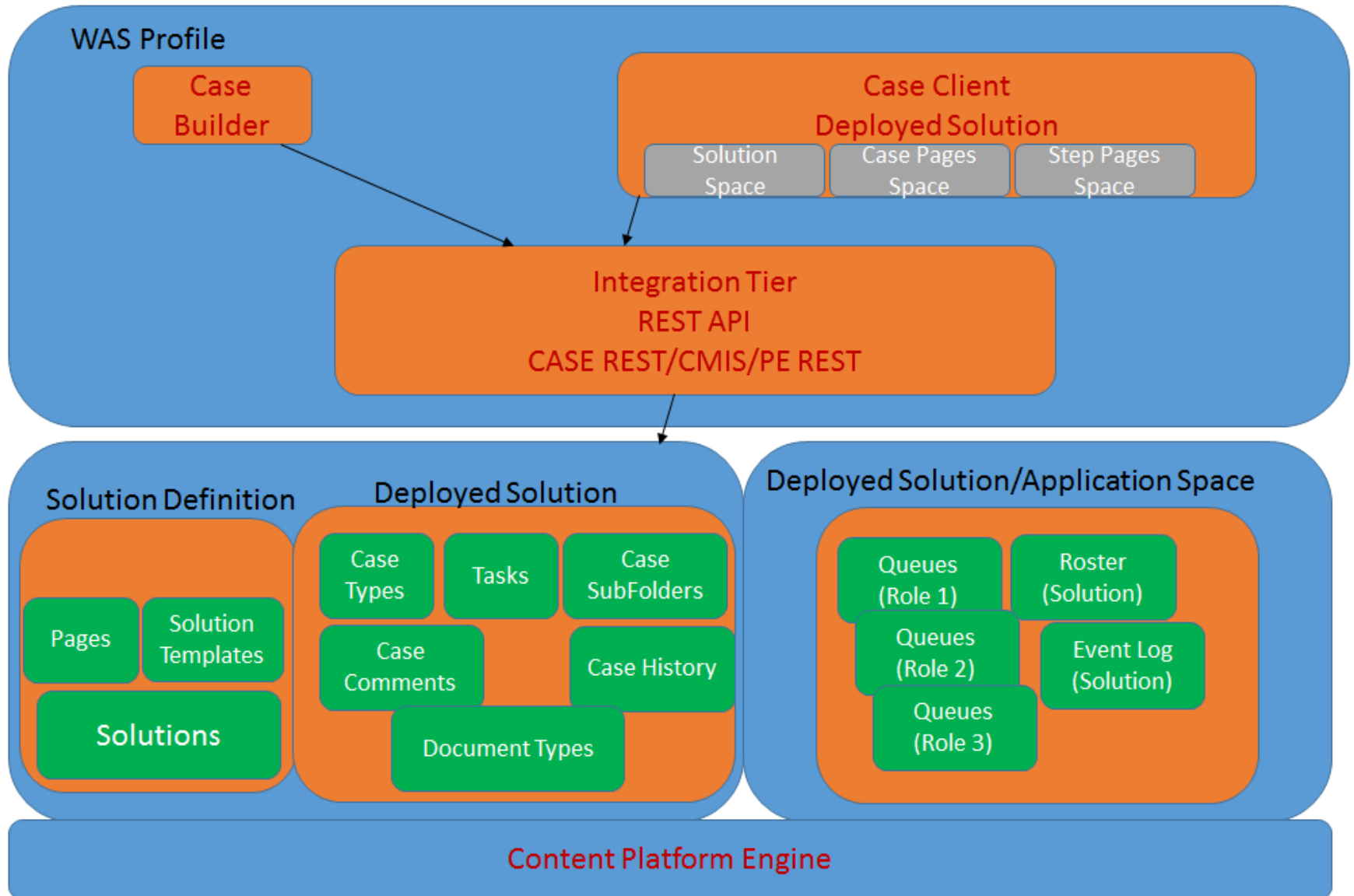
High Level IBM Case Manager Architecture
















How Does a Case Solution Map to a P8 Environment











- IBM Case Manager assets are stored in a associated Content Process Engine Store (Object Store)
 - Associated via PE connection point
- Each object store/isolated region pair can host multiple solutions
 - A solution maps to a PE application space
- In an object store, the solution consists of a set of:
 - Property templates, class definitions, case type classes, task type classes, event actions, subscriptions, event handler code modules
- In an isolated region, the solution consists of a set of:
 - Application spaces, roles, rosters, event logs, work queues, in-baskets
 - The workflow instances and content associated with a case instance have common case and task IDs
- The Case Runtime UI is comprised of a set of spaces and pages hosted in IBM Content Navigator

Object Model – A Detailed View

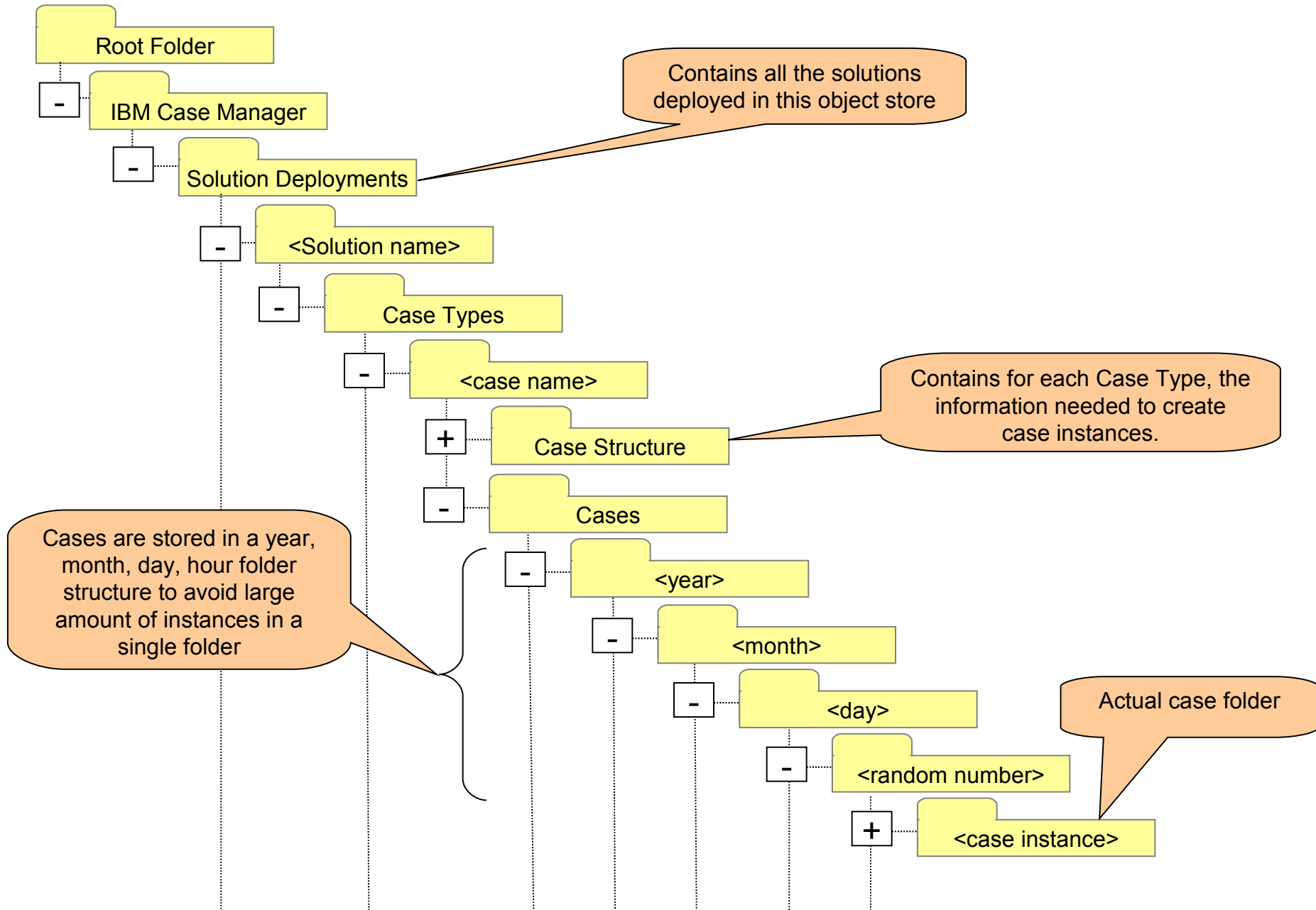


Design Object Store Folder Structure – ICM 5.2

-  Root Folder
 -  CodeModules
 -  IBM Case Manager
 -  Audit Configurations
 -  Connection Definitions
 -  Datasets
 -  Page Templates
 -  Rule Packages
 -  Security Configurations
 -  Solution Templates
 -  Solutions
 -  Widgets
 -  Saved Searches

Containment/Folder Name	Created
 nls	2/20/2014 5:33
 Pages	1/15/2014 4:15
 Views	1/15/2014 4:15
 CCD2_ManageDispute	1/15/2014 4:15
 PE Configuration	1/15/2014 4:15
 Solution Definition	1/15/2014 4:15
 Solution Workflow Collection	1/15/2014 4:15
 CCD2_ManageDispute_Draft	1/15/2014 4:15
 Solution Locks	1/15/2014 4:15
 Solution Workflow Collection_Draft	1/15/2014 4:15

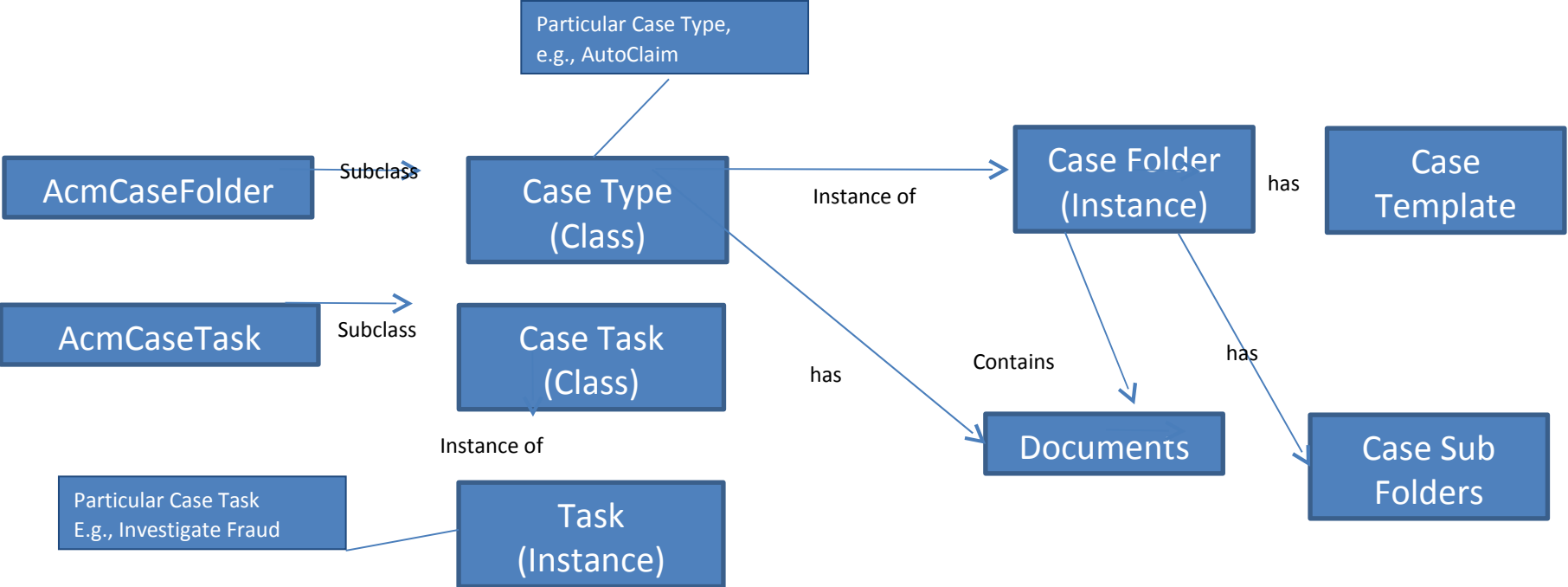
Target object store folder structure



Case Object Model Classes

- Case definitions
 - Symbolic names for all Case Manager based classes and properties begin with the prefix “CmAcm”
 - Deployed solution and cases are managed as folders
 - Folder SubClasses
 - CmAcmBaseCase (abstract)
 - CmAcmCaseFolder (abstract)
 - CmAcmCaseSubfolder
 - CmAcmDeployedSolution
 - CmAcmDeployedCaseType
 - CmAcmCaseTypeSubfolder
 - Case comments are managed as case folder annotations
 - Annotation Subclasses
 - CmAcmCaseComment
 - CmAcmTaskComment
 - CmAcmWorkItemComment
 - CmAcmVersionSeriesComment

CE Case Object Model



Tasks

Case Task

- An activity that must be completed as part of a case
 - Three possible implementations
 - Workflow built from Case Builder in FileNet P8
 - Reuse an existing FileNet BPM workflow
 - Websphere BPM Process
 - Can be in one of these states: Waiting, Ready, Working, Complete, Failed, Disabled
- A task implemented using the Case Builder step editor can be modified using PE Process Designer
 - Each task is a PE workflow
- A task implemented by a WebSphere component must be an empty task in Case Builder (We will focus on the FileNet workflows in this workshop)
- Moving from a formal process-centric to a informal content-centric application model

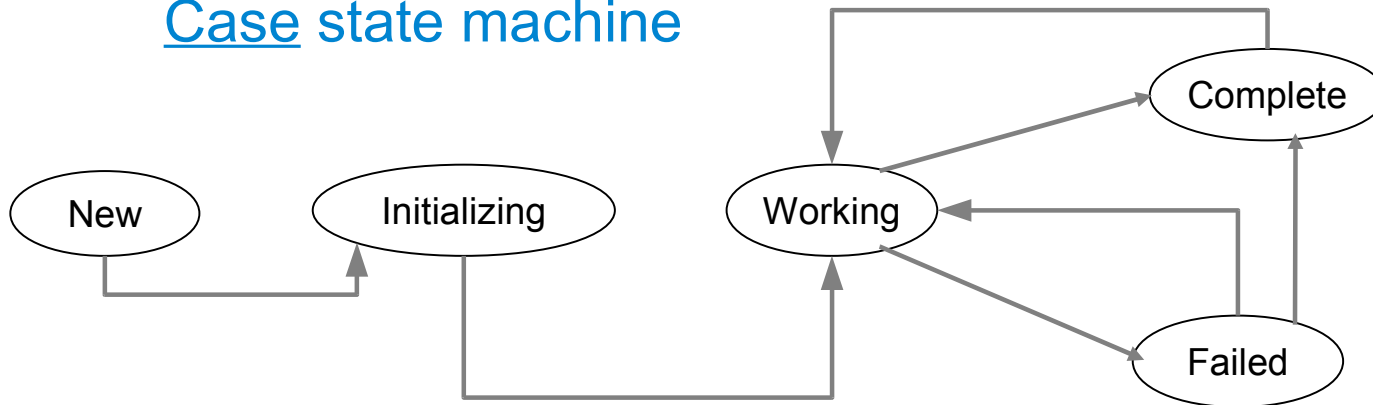
Task Types, States, and Groups

- Types
 - Discretionary (aka Adhoc)
 - Manual
 - Automatic
 - Repeatable
 - Required
 - Optional
 - Custom (aka Dynamic)
- States
 - Ready
 - Waiting
 - Working
 - Complete
 - Failed
 - Disabled
- Groups
 - Ungrouped
 - Mutually Exclusive
 - Inclusive

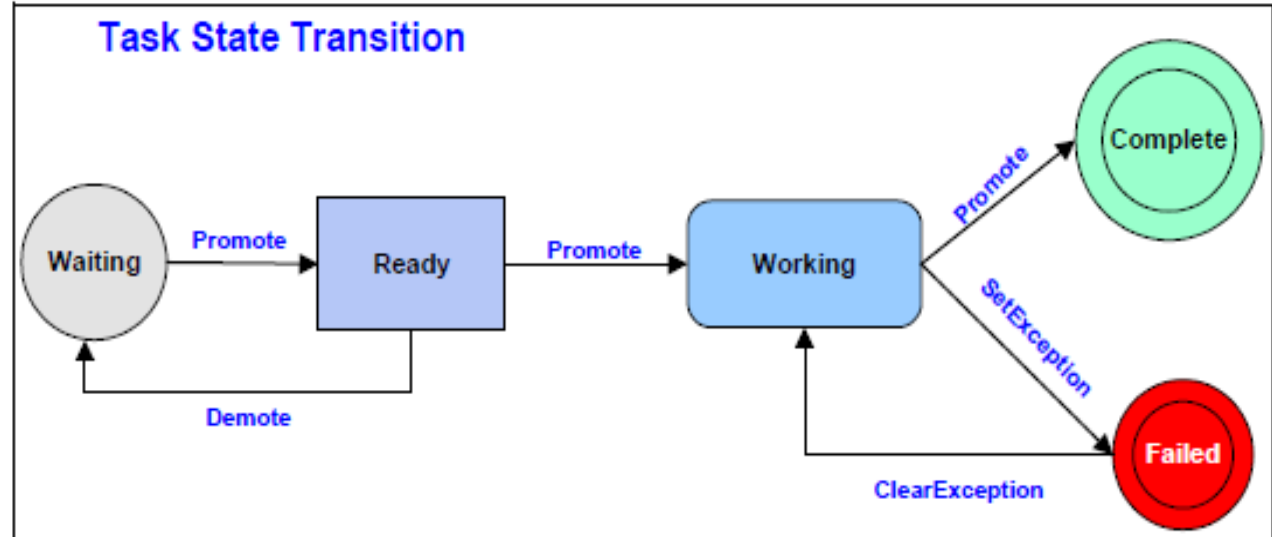
Required versus optional tasks

- A case completes when all required tasks, either
 - reach the completed state
 - are disabled
- Optional tasks are not required for a case to complete

Case state machine



Tasks lifecycle (state machine)



- States
 - Waiting
 - Ready
 - Working
 - Complete
 - Failed
 - Disabled
 - DisabledByExclusive
 - DisabledByUser
 - DisabledAborted

Required Tasks

- Required tasks must be in the following state in order for the case to be completed
 - Automatic & User created:
 - Task must transition to the Complete state or is set to aborted (DisabledState set to DISABLED_ABORTED)
 - Manual
 - The task must either be disabled by the user (DisabledState set to DISABLED_BY_USER), OR
 - Be in the Complete state, OR
 - Set to aborted (DisabledState set to DISABLED_ABORTED)
 - This means the case worker must take action to either start or disable the activity when it is in the Ready state
- A task can become required in the following ways
 - Solution Designer marks the task as required in Case Builder
 - Case Worker creates the task (for Dynamic/Discretionary tasks only), the task is always set to required *
 - Task is part of an inclusive set and a different task in the same set transitioned to Working state
 - Once a task goes into the Working state it becomes required

Manual tasks – starting and disabling

- Manual tasks are started or disabled by case workers
- Case workers can start a manual task once the task enters the Ready state (preconditions met)
- Users can disable a manual task when it is in Ready or Waiting state
- If a Manual task is required, the user must either start the task or disable it in order for the case to complete

Task characteristics

	System created		User created
	Automatic	Manual	
Created when the case is instantiated	Yes	Yes	No
Can have pre-conditions	Yes	Yes	No
Can be marked as 'required' or 'not required' by the Business Analyst	Yes	Yes	No *
Can be manually started by the case worker (i.e. moved from Ready to Working state)	No	Yes	No (automatically starts when created)*
Can be disabled by the case worker when in 'Ready' or 'Waiting' state	No	Yes	No *
Can be re-enabled by the case worker when in Ready state	No	Yes – But only if user disabled – not if system disabled	No *
New task instances can be created by the case worker	No	No	Yes
Can be marked as 'repeatable', where new task instances can be created based on pre-conditions being met more than once	Yes	Yes	No
Can be part of an inclusive group, means system will mark other tasks as required if one in the group moves to working state	Yes	Yes	No
Can be part of an exclusive group, means system will disable other tasks in the group once one moves to the working state	Yes	Yes	No

Task Sets

- Mutually exclusive sets
 - Only one task in the set executes
 - When the first task in the set transitions to execute all other tasks are disabled
- Inclusive sets
 - One task in the set executes, then every single task must execute
 - When the first task in the set transitions to execute all other tasks are marked required

Inclusive and Exclusive Sets

- Automatic and manual tasks that are not repeatable can belong to an inclusive or an exclusive set
 - A task can belong to 0 or 1 set
- Inclusive set
 - If any task in a set transitions to the Working state, then all other tasks in that set become required
 - Examples:
 - Set of tasks related to income replacement; if any of them are needed, then they must all be done
 - Food stamp assessment
 - Child care services assessment
 - Rental assistance assessment
 - Behavior
 - Manual tasks in the inclusive set become required (set to REQUIRED_BY_INCLUSIVE)
 - Case worker must take action to either start the task or disable it
 - Automatic tasks in the inclusive set become required
 - Any optional tasks become required.
 - Note: If the precondition of any required tasks never occurs, then case won't complete.

Inclusive and Exclusive Sets - Continued

- Exclusive set
 - If any task in the set transitions to the Working state, then all other activities in the set become disabled
- Examples:
 - Automatic tasks
 - Provides a mechanism for ‘turning off’ other Automatic tasks in a Waiting state without embedding that information within the precondition
 - If loan over \$2M do a high value review
 - If loan under \$2M do a low value review
 - Manual tasks
 - Provides mechanism for automatically disabling other Manual tasks that might be in the Waiting or Ready state
 - Create a new contract
 - Amend existing contract
- Behavior
 - All tasks except the one that went to Working state will be marked as ‘DISABLED_BY_SYSTEM’.
 - A case worker cannot re-enable a manual task that was disabled by the system as the result of being in an Exclusive set

Repeatable Tasks

- Automatic and Manual tasks can be identified as repeatable in Case Builder
- The first instance of a repeatable task is created when the case instance is created
- A repeatable task must have a pre-condition
- New tasks are created when preconditions are met more than once
 - the new, repeated task is created automatically by the system
- A user-created task cannot be marked as 'repeatable'
- It is not possible to create another instance of a repeatable task from the Case Client
- Repeatable tasks cannot be in an inclusive or exclusive group
- What happens when a repeatable manual task is disabled?
 - It stays disabled, it has no impact on new repeatable tasks that are created as the result of a precondition
- Examples
 - Police report
 - Witness statements (automatic) – 0..n witness statements may arrive
 - Resume
 - When a new resume is filed in the case (which is a position being filled), a new task for reviewing the resume is created

Failures and Aborting Tasks

- When an exception has occurred for a task that is in the Working state, the task is set to FAILED state
 - Example: If an exception occurs in the workflow
 - Exceptions may be recoverable, an administrator/tracker must determine the course of action
 - A Case cannot be completed when a task is in the Failed state, unless the task has been Aborted
 - Error message text is recorded in the Task object
- An Aborted task provides a mechanism for indicating that a task is no longer needed, and that the case can complete without the task reaching the COMPLETED state
 - Administrators would explicitly set Abort using ACM administration tools
 - PE Administrators can abort a task by terminating the process associated with the task. This is done using Process Administrator
 - A task that has been aborted will have DisabledState set to DISABLED_ABORTED
- Abort and Failed states cannot be corrected from the Widget UI, however, the user can see the Abort and Failed status

IBM Case Manager Builder

P8Admin ? IBM

Manage Solutions \ Credit Card ... \ Manage Dispute

Show Locked Items Back Validate Save Save and Close

Case Type

Properties

Views

Case Folders

Rules

Tasks

?

Add Task

Manage Sets

All tasks

View by: Priority | Set | Name

Required tasks

Review Dispute Item

Precondition:
Case Start

Set: <None>

Optional tasks

Generate Correspond...

Precondition:
Case Start

Set: <None>

Discretionary tasks

Create Review Request

Set: <None>

27

Cases and Tasks

© 2011 IBM Corporation

Required tasks

Review Dispute Item

Precondition:
Case Start

Set: <None>

Optional tasks

Generate Correspond...

Precondition:
Case Start

Set: <None>

Discretionary tasks

Create Review Request

Set: <None>

General

Preconditions

Task Properties

Design Comment

* Name:

Review Dispute Item

* Unique Identifier

CCD2_ReviewDisputeItem

Description:

This task starts:

☒ Automatically
 ☐ Manually
 ☐ Discretionally

This task is:

☐ Hidden
 ☒ Required
 ☐ Automatically completed when the case completes

Assign to set:

<None>

Manage Sets

OK

Cancel

The difference between custom tasks and ad hoc tasks

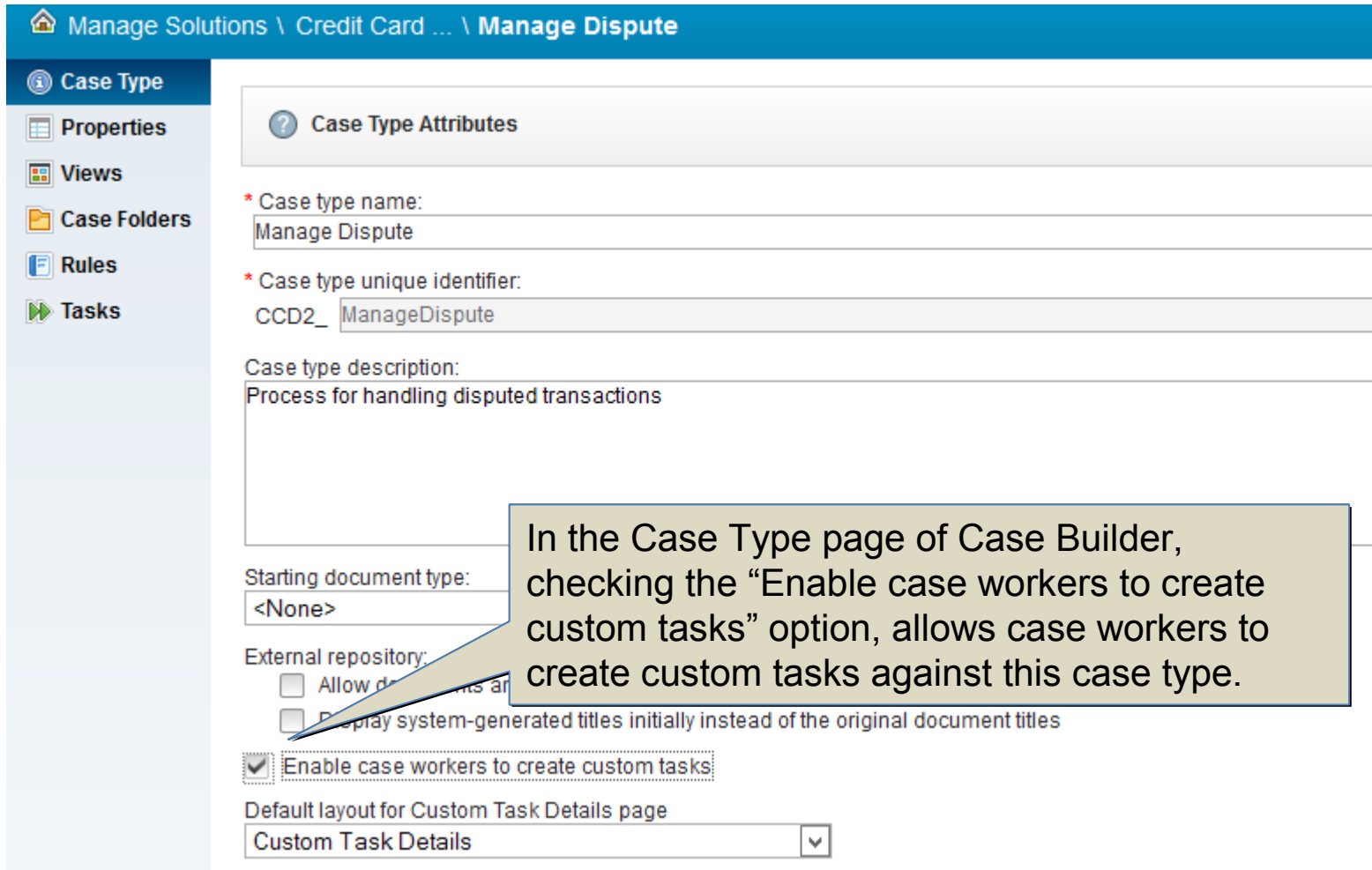


- Ad hoc task is designed with IBM Case Builder at design-time by business analyst
- Custom task is designed with IBM Case Manager Client at run-time by case worker

When to use ad hoc vs. custom tasks

- *Ad hoc tasks are predefined with IBM Case Builder, they are reused by case workers on a regular basis, so they are likely to be used by case workers working on different cases*
- *Custom tasks should be reserved for unexpected tasks that the case worker needs to complete in order to be able to process their assignments, this kind of task can be designed simply, they are likely to be only used for specific cases*
- *If you see similar custom tasks being created on a regular basis, or a specific custom task being used frequently, consider whether you should create an ad-hoc task that can be used instead of the custom task*
- *When a large number of custom tasks are used, or when a specific custom task is used extensively, it is likely that the case definition logic is deficient in some way and should be re-evaluated.*

Enabling and using custom tasks (1)



The screenshot shows the 'Manage Dispute' page in the Case Builder application. The left sidebar contains a navigation menu with 'Case Type' selected, and sub-items: 'Properties', 'Views', 'Case Folders', 'Rules', and 'Tasks'. The main content area is titled 'Case Type Attributes' and contains the following fields:

- * Case type name:** Manage Dispute
- * Case type unique identifier:** CCD2_ManageDispute
- Case type description:** Process for handling disputed transactions
- Starting document type:** <None>
- External repository:**
 - ☐ Allow documents to be added to the external repository
 - ☐ Display system-generated titles initially instead of the original document titles
- ☒ **Enable case workers to create custom tasks:**
- Default layout for Custom Task Details page:** Custom Task Details

A callout box points to the 'Enable case workers to create custom tasks' checkbox with the following text:

In the Case Type page of Case Builder, checking the “Enable case workers to create custom tasks” option, allows case workers to create custom tasks against this case type.

Enabling and using custom tasks (2)

- Case workers who need to generate a custom task
 - Log on IBM Case Manager Client

Search: Case Identifier % Search Advanced Search

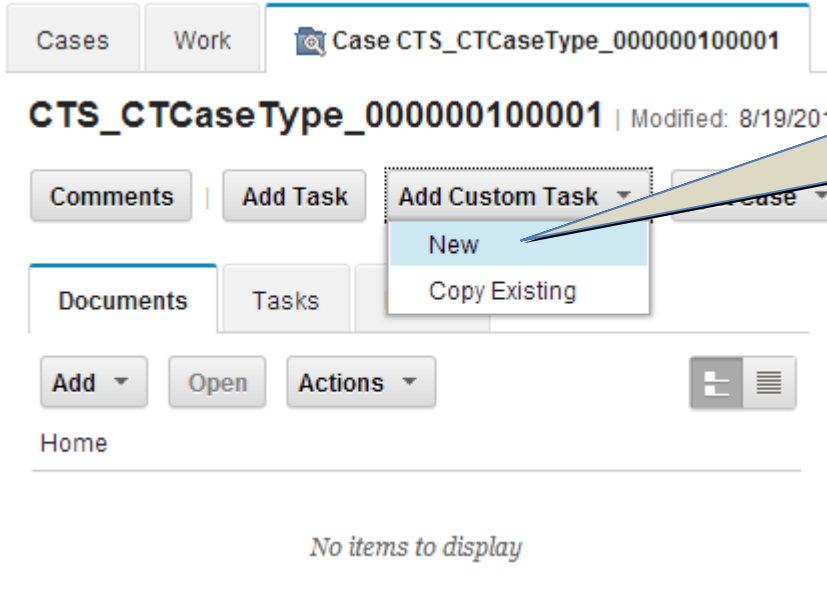
Title	Status
CTS_CTCaseType_000000100001	Working
CTS_CTCaseType_000000110001	Working

Open Case
Show Link to Case
Send Link to Case
Comments

Selecting a case and clicking "Open Case" to open "Case Details" page for this case.

In the "Cases" page, clicking the "Search" button to search cases.

Enabling and using custom tasks (3)



The screenshot shows the 'Case Details' page for 'Case CTS_CTCaseType_000000100001'. The page has tabs for 'Cases' and 'Work'. Below the case title, there are buttons for 'Comments', 'Add Task', and 'Add Custom Task'. The 'Add Custom Task' dropdown menu is open, showing options 'New' and 'Copy Existing'. A callout box points to the 'New' option with the text: 'In the “Case Details” page, clicking “Add Custom Task” -> “New” to open “Custom Task Editor” dialog.' Below the dropdown, there are sections for 'Documents' and 'Tasks'. The 'Tasks' section shows 'Add', 'Open', and 'Actions' buttons. The 'Documents' section shows 'Add', 'Open', and 'Actions' buttons. The 'Tasks' section also shows a 'String' input field. The 'Documents' section shows a 'String' input field. The 'Tasks' section shows 'No items to display'.

In the “Case Details” page, clicking “Add Custom Task” -> “New” to open “Custom Task Editor” dialog.

Enabling and using custom tasks (4)

The screenshot shows the 'Custom Task Editor' window. At the top, there are three buttons: 'Validate', 'Save', and 'Start'. Below these is a text input field with the placeholder text 'Enter a name for the task'. Underneath the text field is a rich text editor toolbar with icons for undo, redo, bold, italic, underline, and a list of task types. The task name 'Fraud Detection' is entered, followed by a description 'To do fraud detection'. Below the description, it says 'Work items for this task:' and there is a link 'Insert an item'. A dropdown menu is open, showing three options: 'Work item' (with a person icon), 'Decision point' (with a diamond and arrows icon), and 'Wait until or for' (with a clock icon). Callouts point to each of these elements with explanatory text.

Custom Task Editor

Validate Save Start

Enter a name for the task

Fraud Detection

To do fraud detection

Enter a description of the task

Work items for this task:

[Insert an item](#)

Work item

Decision point

Wait until or for

Create a person-activity work item

Create a decision point for a person-activity work item

This kind of step is used to indicate how long the task flow waits before continuing to execute



Enabling and using custom tasks (5)

Work items for this task:

1  **Wait Until** 

Wait **Until**

Case workers can design “Wait Until” steps which indicate date-time the task flow waits before continuing to execute

2  **Wait For** 

Wait **For**

[Insert an item](#)

Hours
Days
Weeks

Case workers can design “Wait For” steps which indicate Hours/Days/Weeks the task flow waits before continuing to execute

Enabling and using custom tasks (6)

The screenshot displays the IBM Case Manager interface for a task titled "Fraud Detection". The task description is "To do fraud detection". Below this, a section titled "Work items for this task:" lists several work items. The first work item is "1 Follow up claim", assigned to "P8 Administrator". A callout box points to this assignment with the text: "Case workers can assign specific person or role to a work item. If assigning a role, everyone in that role can process the work item." Below the first work item is a "Decision point:" section. It contains two main responses: "2.1 Response: Approve" and "2.2 Response: Start fraud detection". Under "2.1 Response: Approve" is a sub-work item "2.1.1 Process Claim" assigned to "P8 Administrator". Under "2.2 Response: Start fraud detection" is a sub-work item "2.2.1 Detect Fraud" assigned to "Claimer". Each work item has a link to "Insert an item" and a placeholder text "[Click to add instructions for this work item]". The interface includes a rich text editor toolbar at the top with various icons for text formatting and document management.

Enabling and using custom tasks (7)

The screenshot displays the IBM Work Item editor interface. At the top, there are buttons for 'Validate', 'Save', 'Start', and 'Close'. Below these is a rich text toolbar with icons for undo, redo, bold, italic, underline, strikethrough, bulleted list, numbered list, link, unlink, and other text formatting options. The main content area shows a task titled 'Fraud Detection' with the instruction 'To do fraud detection'. Below this, it lists 'Work items for this task:' with one item: '1 Process claim' assigned to 'Fraud Team'. The instruction for this work item is 'Please take a look at [Penguins.jpg](#) and [Modified By](#), then connect to [IBM](#).' At the bottom, there is a link 'Insert an item'.

Callouts provide the following information:

- The instruction for custom task:** Points to the 'To do fraud detection' text.
- Case document link button:** Points to the document icon in the toolbar.
- Case property link button:** Points to the list icon in the toolbar.
- Url link button:** Points to the globe icon in the toolbar.
- Click "Deadline" button to set deadline and reminder for work item:** Points to the calendar icon in the toolbar.
- Click "Editable properties" button to add editable case properties for work item:** Points to the icon with a pencil and checkmark in the toolbar.
- The instruction for work item. Case worker can also click buttons on the top toolbar to add case document link, case property link, and url link here, then the instruction will be displayed in "work detail" page:** Points to the work item instruction text.

Enabling and using custom tasks (8)

- Case workers receive the work item and open “Work Details” page for this work item

The instruction for work item is displayed here

Process claim *

Comments

Instructions

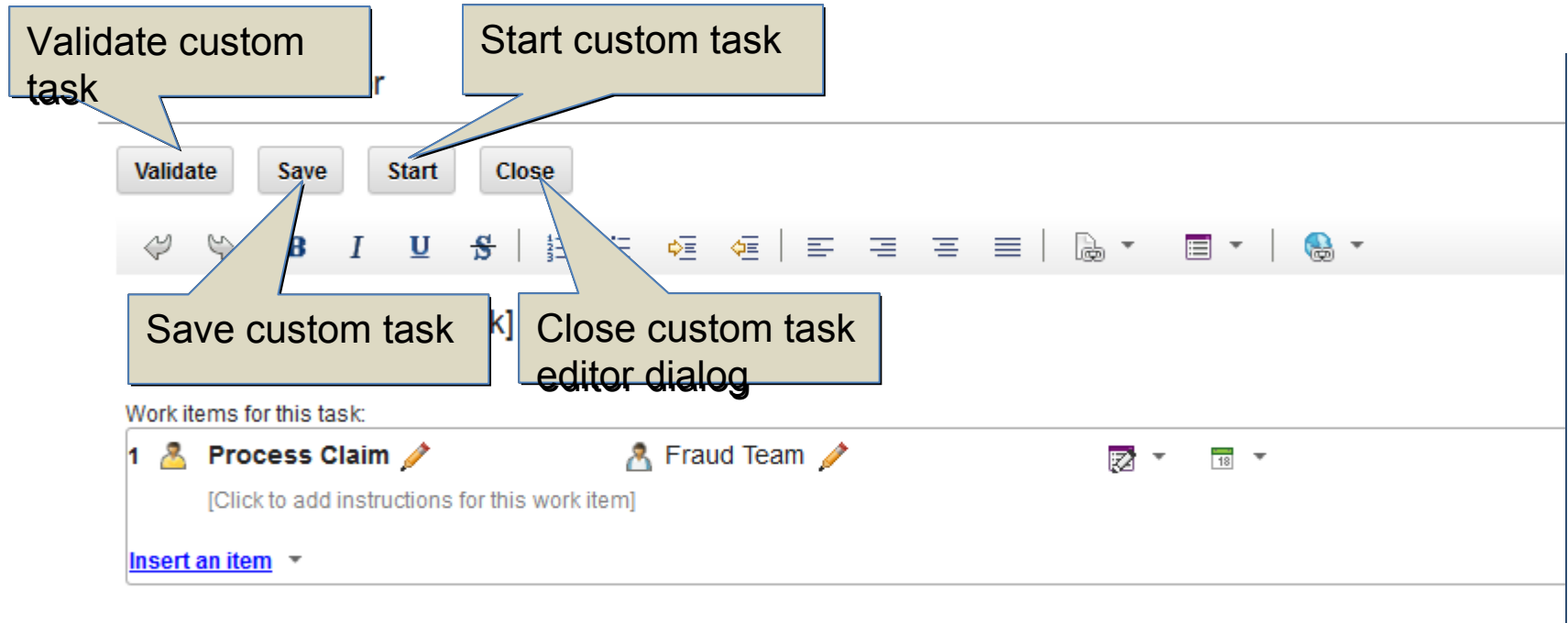
Please take a look at [Penguins.jpg](#) and Modified By: P8 Administrator, then connect to [IBM](#).

String

Editable property here

Editable properties for work item are displayed with “Properties” widget, case workers can edit property value here

Enabling and using custom tasks (9)



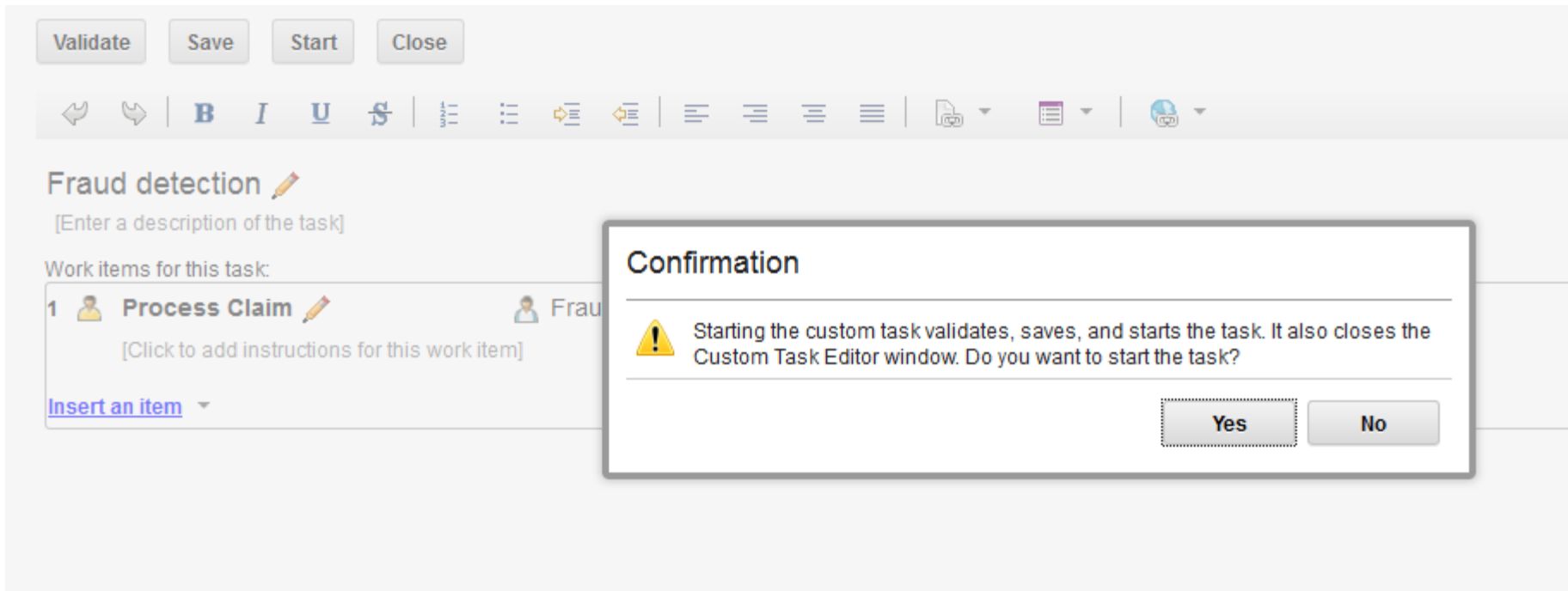
Enabling and using custom tasks (10)

- Case workers should validate custom tasks before saving and starting
- If case workers intend to save or start custom tasks which are invalid, a validation errors dialog will popup

The screenshot shows a web-based task creation interface. At the top, there are four buttons: 'Validate', 'Save', 'Start', and 'Close'. Below these is a rich text editor toolbar with icons for undo, redo, bold, italic, underline, strikethrough, bulleted list, numbered list, indent, outdent, link, unlink, and a help icon. The main form has two text input fields: '[Enter a name for the task]' and '[Enter a description of the task]'. Below these is a section titled 'Work items for this task:' containing a list item '1 Process Claim' with a user icon and an edit icon. A link '[Click to add instructions for this work item]' is below the list item. At the bottom left, there is a link 'Insert an item' with a dropdown arrow. A 'Validation errors' dialog box is open in the foreground, displaying an information icon and the message 'The task has the following errors:'. Below this, a list contains one error: '1. The task has no name.'. An 'OK' button is at the bottom right of the dialog.

Enabling and using custom tasks (11)

- If case workers click “Start” before “Save”, it validates, saves, and starts the task. It also closes the custom task editor window.



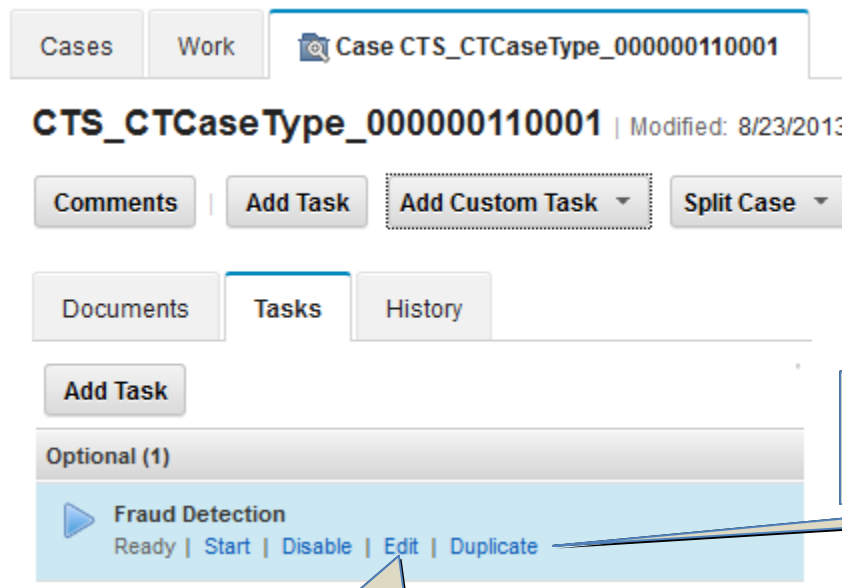
Enabling and using custom tasks (12)

- If case workers click “Close” before “Save” or “Start”, they are prompted to discard or save the task

The screenshot displays a web-based task management interface. At the top, there are four buttons: 'Validate', 'Save', 'Start', and 'Close'. Below these is a rich text editor toolbar with icons for undo, redo, bold, italic, underline, strikethrough, bulleted list, numbered list, link, unlink, indent, outdent, decrease indent, increase indent, and a dropdown menu. The main content area shows a task titled 'Fraud detection' with a pencil icon and a placeholder text '[Enter a description of the task]'. Below this, it says 'Work items for this task:' and lists one item: '1 [person icon] Process Claim [pencil icon]' with the assignee 'Fraud Team' and a placeholder '[Click to add instructions for this work item]'. At the bottom left, there is a link 'Insert an item' with a dropdown arrow. Overlaid on the right side of the interface is a modal dialog box titled 'Unsaved changes'. It contains a yellow warning triangle icon and the text 'You have unsaved changes. Do you want to save the changes?'. At the bottom of the dialog are three buttons: 'Discard' (highlighted with a dashed border), 'Save', and 'Cancel'.

Enabling and using custom tasks (13)

- If case workers save custom tasks, they can be found again in “Case Details” page



Case workers can duplicate custom tasks whether or not they are started














Case workers can edit custom tasks if they are not started.

Enabling and using custom tasks (14)

Fraud Detection

To do fraud detection

Work items for this task:

- 1  **Investigate**  Correspondence Team
[Click to add instructions for this work item]
- 2  **Decision point:**
 - 2.1  **Response:**
 - 2.1.1  **Process claim**  Dispute Advisor
[Click to add instructions for this work item]
 - 2.2  **Response:**
 - 2.2.1  **Start fraud detection**  Fraud Analyst
[Click to add instructions for this work item]
- 3  **Conclusion**   Data Clerk 
[Click to add instructions for this work item]

When custom task is started, “Investigate” work item will be sent to “Correspondence Team”, if someone chooses “Approve”, “Process claim” work item will be sent to “Dispute Advisor”, If someone chooses “Reject”, “Start fraud detection” work item will be sent to “Fraud Analyst”, finally, “Data Clerk” will receive “Conclusion” work item.

Task Enhancements

- ➔ Task Type and Preconditions
 - Container Task & Subtask
 - Hidden Task
 - Task Stop & Restart
 - Course Summary

Background

- *Task Type can be*
 - *Automatic, Manual, Discretionary*
 - *Discretionary used to be called User Creatable*
 - *It can be created either by the user or via Case REST/Java API*
 - *Required or Not Required*
 - *Made to start only after Preconditions are met*
- *Customers and partners requested support for*
 - *Better monitoring of changes in case property values*
 - *More flexibility in building combined preconditions*
 - *Allowing changes in task type and preconditions while redeploying the solution*

Task Preconditions

- *No precondition*
- *When property condition is met*
 - *A property criteria expression only*
 - e.g. *(Priority = high AND LoanAmount >= 100000)*
- *Property update precondition*
- *Document filing precondition*

Task Preconditions (Revised Case Builder UI)

- No precondition

General

Preconditions

Design Comment

What preconditions must be met for this task to start?

No preconditions

▼

- A property condition is met

General

Preconditions

Design Comment

What preconditions must be met for this task to start?

A property condition is met

▼

Add Condition

Delete All Conditions

Match:

All

All

Any

Property	Operator	Value	
Dispute Case State	=	Arbitrate	AND
Disputed Amount	>=	5,000	

Property Update Precondition

- *When a property (or properties) is changed*
 - e.g. (CaseProp1 OR CaseProp2) changed
 - Can be AND'ed with property criteria expression as well, e.g.
 - (CaseP1 OR CaseP2) changed
 - AND (DollarAmount >= 10000 AND Status=true)
- *Repeatable*
 - When the event happens, if there is no existing task of that type in the waiting state
 - A new task is created by the system
 - Task then evaluated against its precondition

Property Update Precondition (New Case Builder UI)

- A case property is updated

GeneralPreconditionsDesign Comment

What preconditions must be met for this task to start?

A case property is updated

☒ Task is repeatable

Case properties:

Date Submitted

Dispute Case State

Dispute Description

Dispute Type

Disputed Amount

The above precondition and the following conditions:

Add ConditionDelete All Conditions

Match: All

Property	Operator	Value
Dispute Case State	=	Arbitrate

Document Filing Precondition

- *Allows any document type, or select 1 or more document types, e.g.*
 - *Any document that is of base Document class*
 - *(DocType1 OR DocType2) filed in the case*
- *Can be AND'ed with property criteria as well, e.g.*
 - *(DocType1 OR DocType2) filed in the case*
 - *AND (DollarAmount >= 10000 AND Status=true)*
- *Repeatable*

Document Filing Precondition (Revised Case Builder UI)

- A document is filed in the case*

General
Preconditions
Design Comment

What preconditions must be met for this task to start?

A document is filed in the case

☒ Task is repeatable

☐ Any document type

Document types:

Dispute Form
Phone Recording
Policy Report
Supporting Document
Survey Form

The above precondition and the following conditions:

Add Condition

Delete All Conditions

Match: All

Property	Operator	Value
Disputed Amount	>=	5000

Task Precondition Evaluation

- *When a case is created*
 - *Non-discretionary top level tasks are created and put in a waiting state*
 - *Tasks with no precondition or with property conditions that are met*
 - *Are moved to ready state*
 - *If automatic, the task moves to working and the associated workflow launches*
 - *Tasks with property update or document filing conditions*
 - *Wait for the property update or document filing event to happen*
If property criteria exists but does not evaluate to a successful result (true), the task stays in the waiting state until the event trigger reoccurs

Task Type Redeployment

- *Run the Precondition Checker after redeploying the solution in the production system*
- *Change from Automatic/Manual to Discretionary not recommended for production systems*
 - *The characteristics of task process launching via system subscription vs. user involvement with Launch UI are quite different and can cause incompatibility with existing task instances*

Precondition Checker

- *Mostly for production systems to keep tasks updated/processed and keep up with changing business requirements*
- *After a solution is deployed, if a precondition criteria/expression has been changed for “A property condition is met” task type, run the Precondition Checker to check for existing task instances that are in the waiting state as they may now satisfy the new precondition criteria/expression*
- *For case property updates and document filing trigger task types, the precondition checker does not need to be run even if the combined property criteria/expression is changed*
 - *Wait for the next case property update or document filing to trigger the precondition evaluation automatically, which will use any new property criteria/expression accordingly*
 - *There is no detection of document already filed/exists, may need to unfile/file again as needed*

Precondition Checker (cont.)

- *Also run the Precondition Checker if Task Type precondition type changes*
 - *From “A case property is updated” or “A document filed in the case” to “A property condition is met”*
 - *From any kind of precondition and change to “No precondition”*

in order to validate all existing task instances after redeploying a solution

- *Precondition Checker is a command line utility*
 - *Refer to ICM Info Center for command line details and options*

Task Enhancements

- Task Type and Preconditions
- ➔ Container Task & Subtask
- Hidden Task
- Task Stop & Restart
- Course Summary

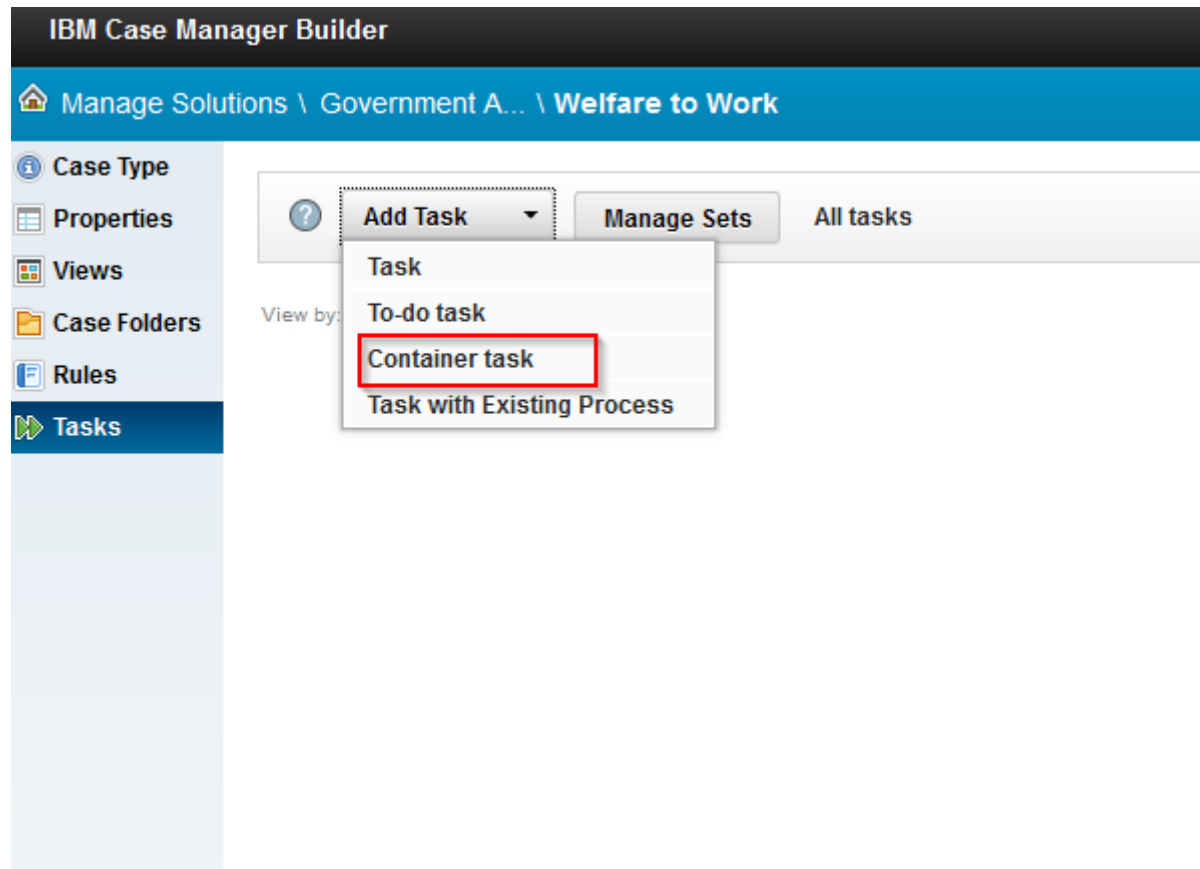
Background – Container Task and Subtask

- *Help manage complexity and organize tasks logically and structurally in large case management solutions*
 - *As cases reach certain phases, logically makes additional tasks available in stages*
 - *Design high level tasks first, then break down each task into smaller subtasks*
- *Not all non-discretionary tasks need to be instantiated when a case is created*
 - *More efficient case creation*
 - *Better use of CE task objects when needed*
 - *Less complexity in Case Client UI for case workers*

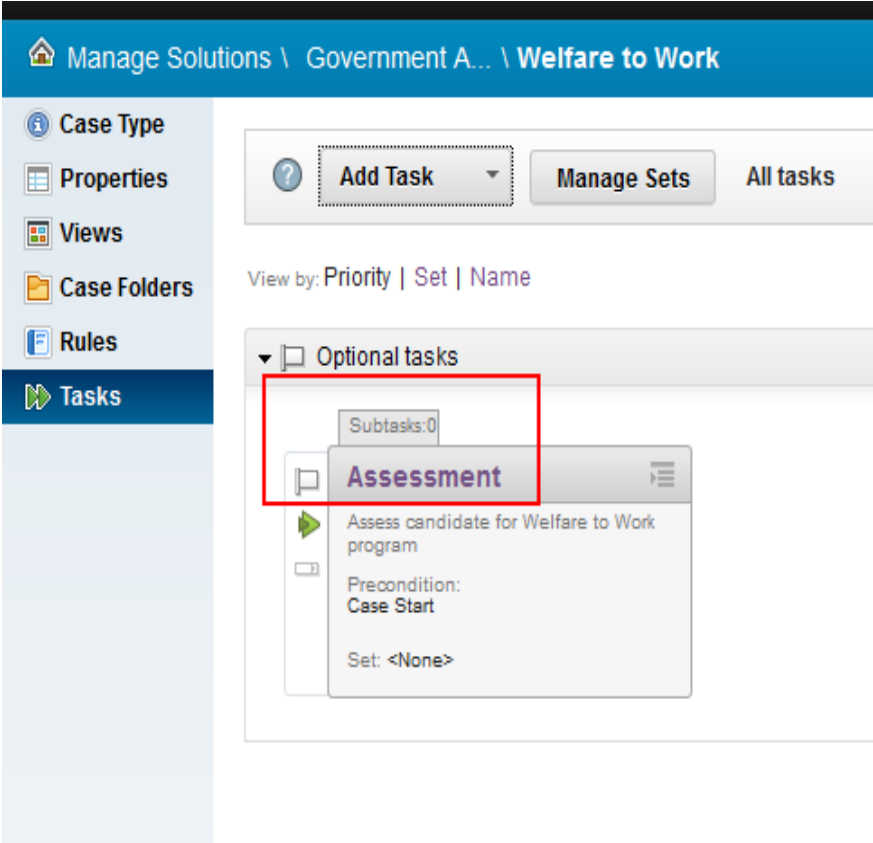
Container Task

- *Does not have workflow process associated with it*
- *Contains subtasks*
- *Supports features like regular task type*
 - *Preconditions*
 - *Automatic/Manual/Discretionary*
 - *Required or not*
 - *Hidden or not*
 - *Can participate in inclusive/exclusive set*

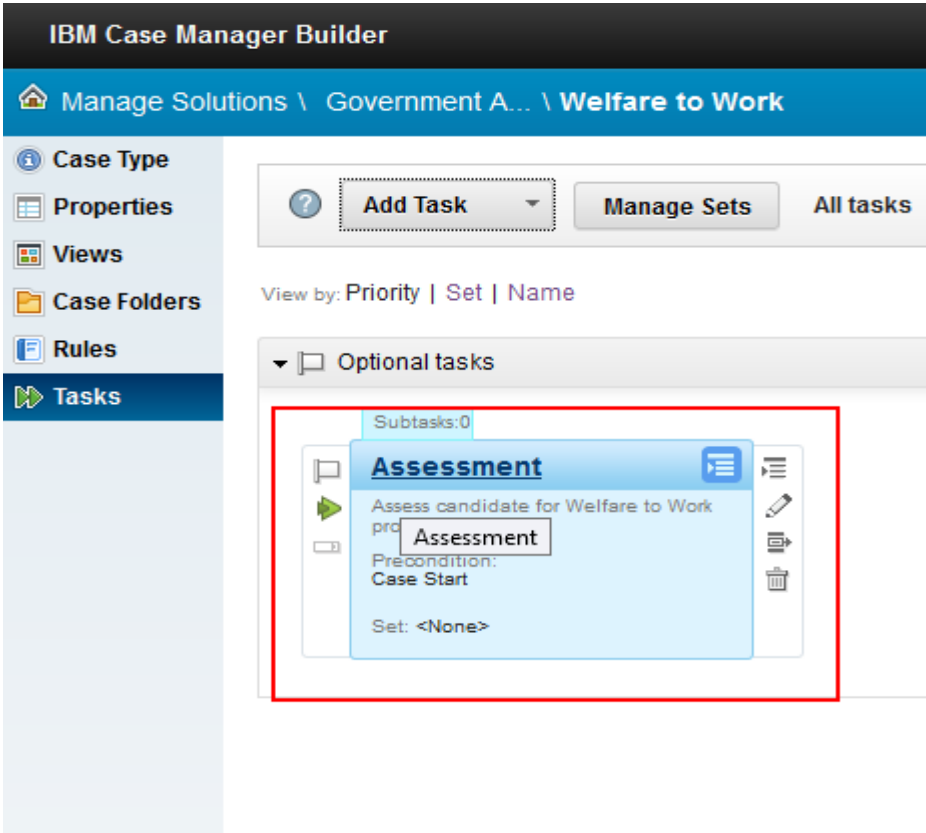
Case Builder UI to Create Container Task



Container Task



View without Mouse Hover

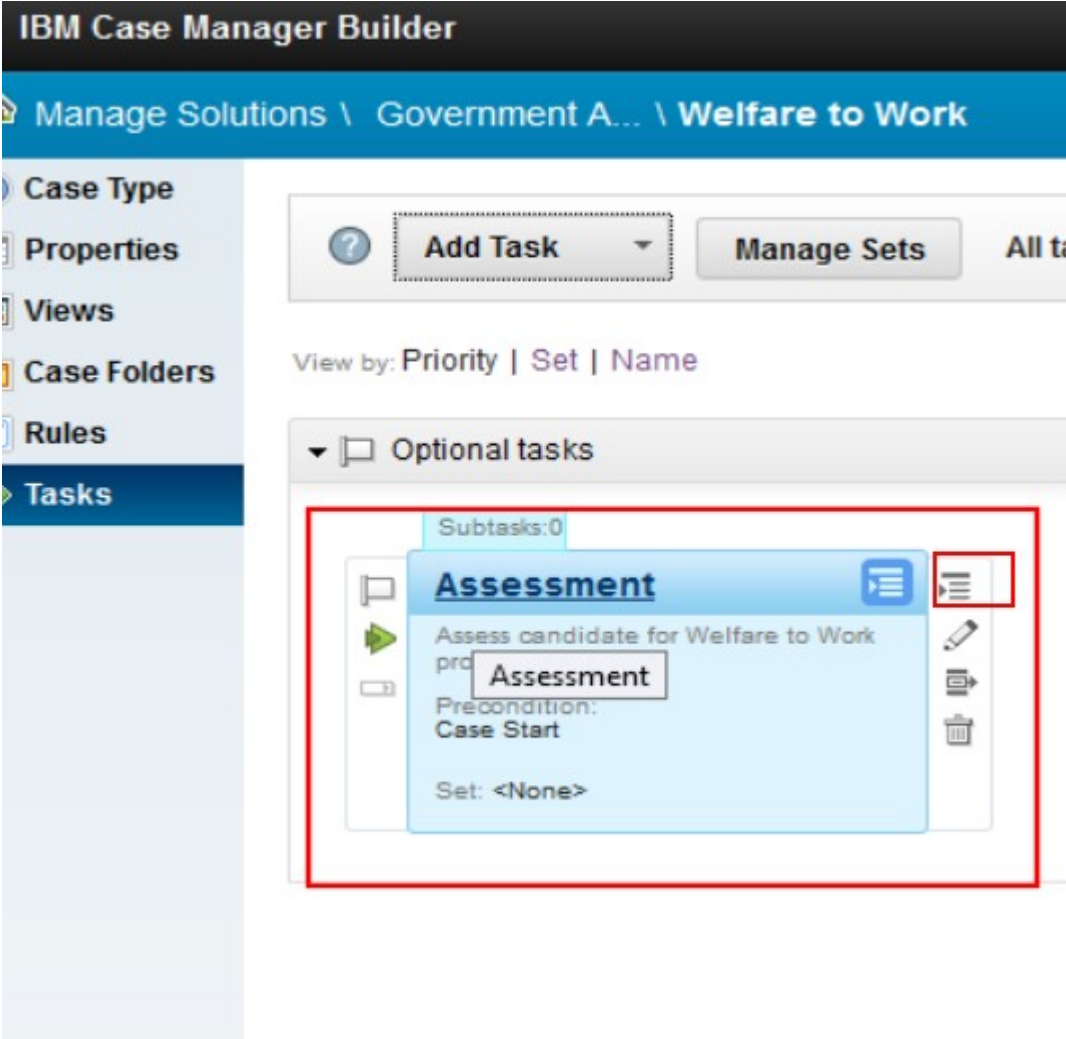


View with Mouse Hover

Subtask

- *Subtasks are created inside a container task.*
- *Subtasks can also be container tasks or regular tasks*
- *Subtasks can have*
 - *Preconditions*
 - *Automatic/manual, but not discretionary*
 - *Required or not*
 - *Hidden or not*
 - *Can participate in inclusive/exclusive set*
 - *All subtasks in the set have to be subtasks of the same container task, i.e. inclusive/exclusive set can not cross container task boundaries*

Subtask



Case Builder UI to Create Subtask

IBM Case Manager Builder

Manage Solutions \ Government A... \ Welfare to Work

Case Type
Properties
Views
Case Folders
Rules
Tasks

View by: **Add Subtask** Manage Sets All tasks \ xxx

Subtask
To-do task
Container subtask
Subtask with Existing Process

Add a subtask

General Preconditions Task Properties Design Comment

* Name:

* Unique Identifier
GA_

Description:

This task starts:
☒ Automatically ☐ Manually

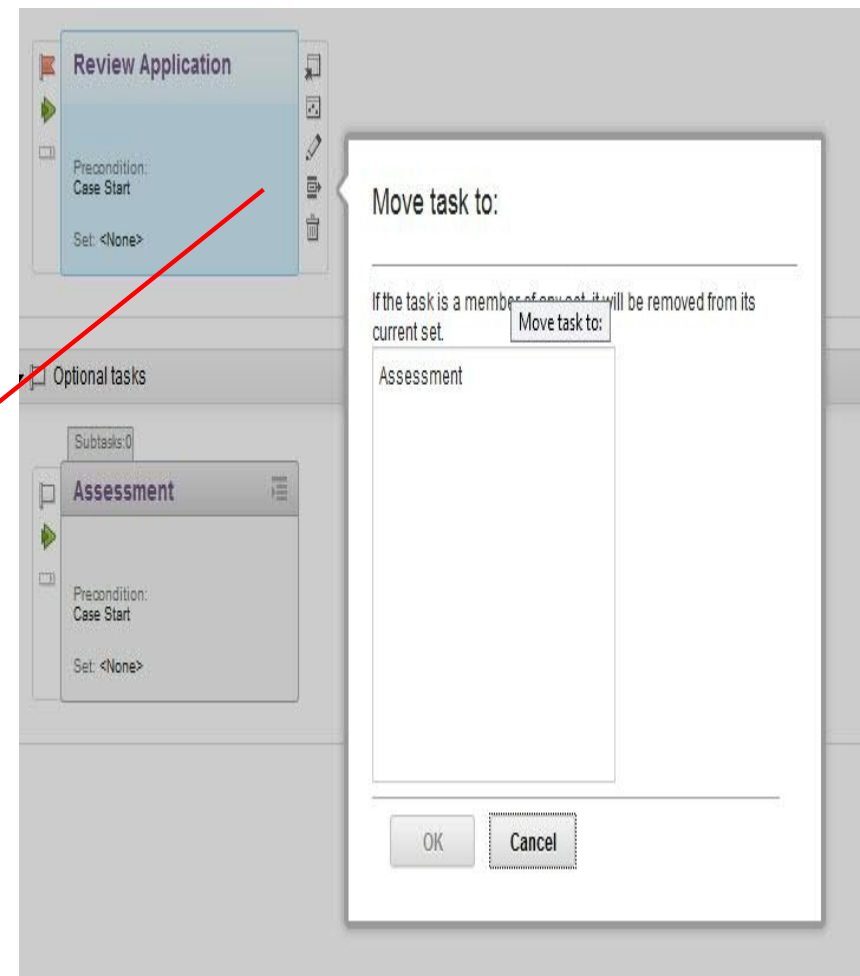
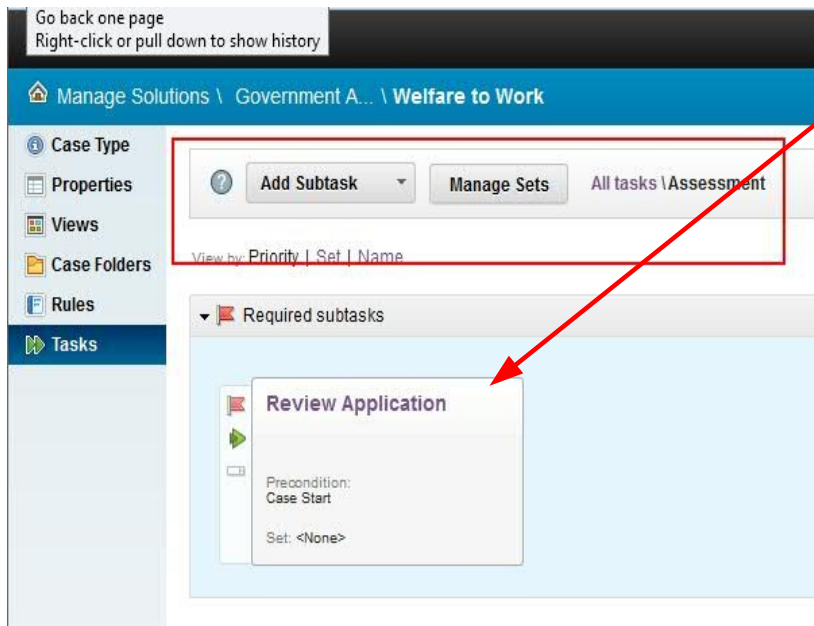
This task is:
☐ Hidden
☐ Required

Assign to set:
<None>

OK Cancel

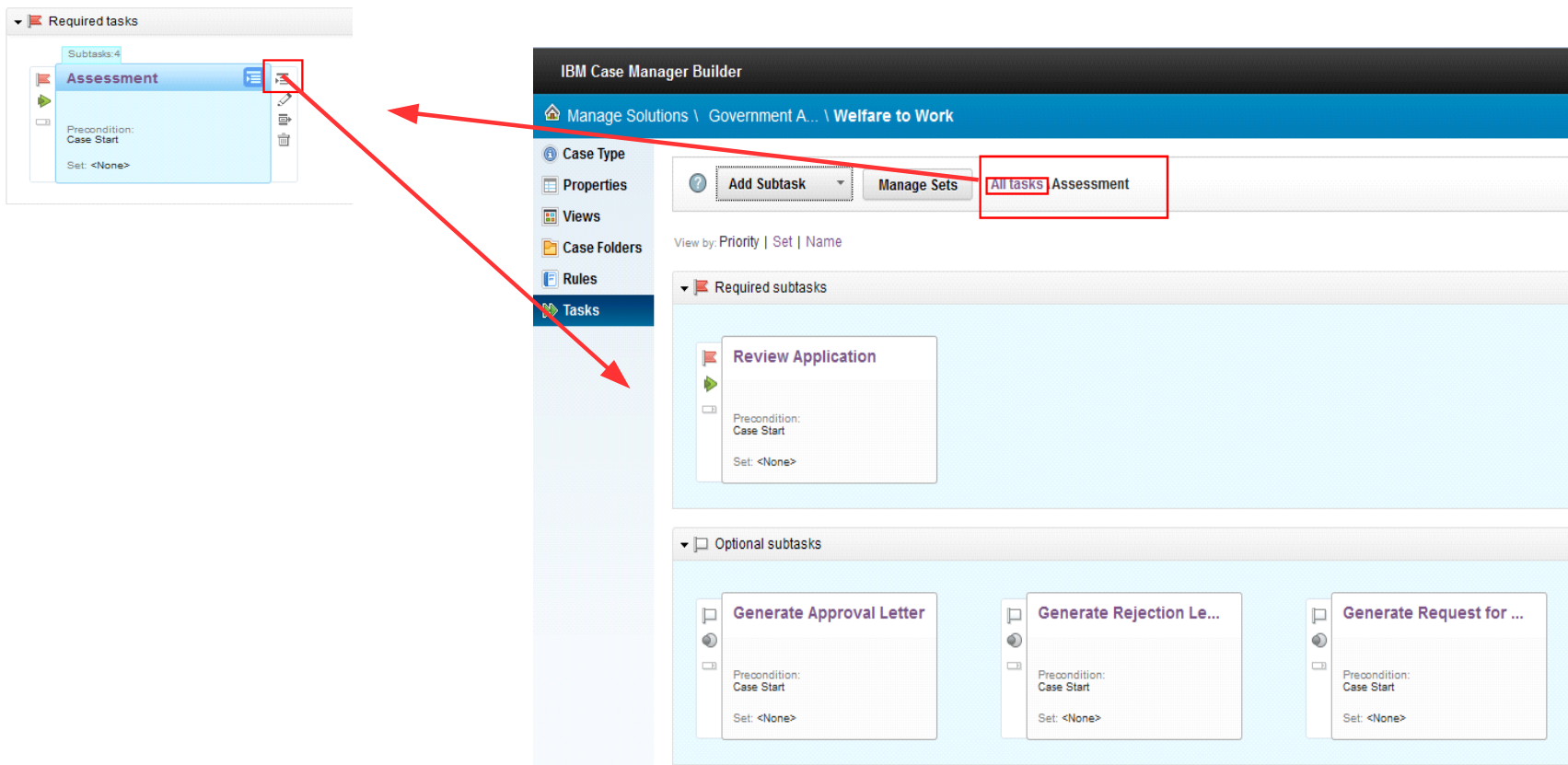
Move Existing Task to Container Task

- *Structurally organize existing Tasks into new Container Tasks*
- *Optionally, add Container Task precondition to group subtasks logically*



Display and Navigate Container Task Type

- *Different color representation in Case Builder UI*
- *Navigate into Container Task and out with container path breadcrumb*

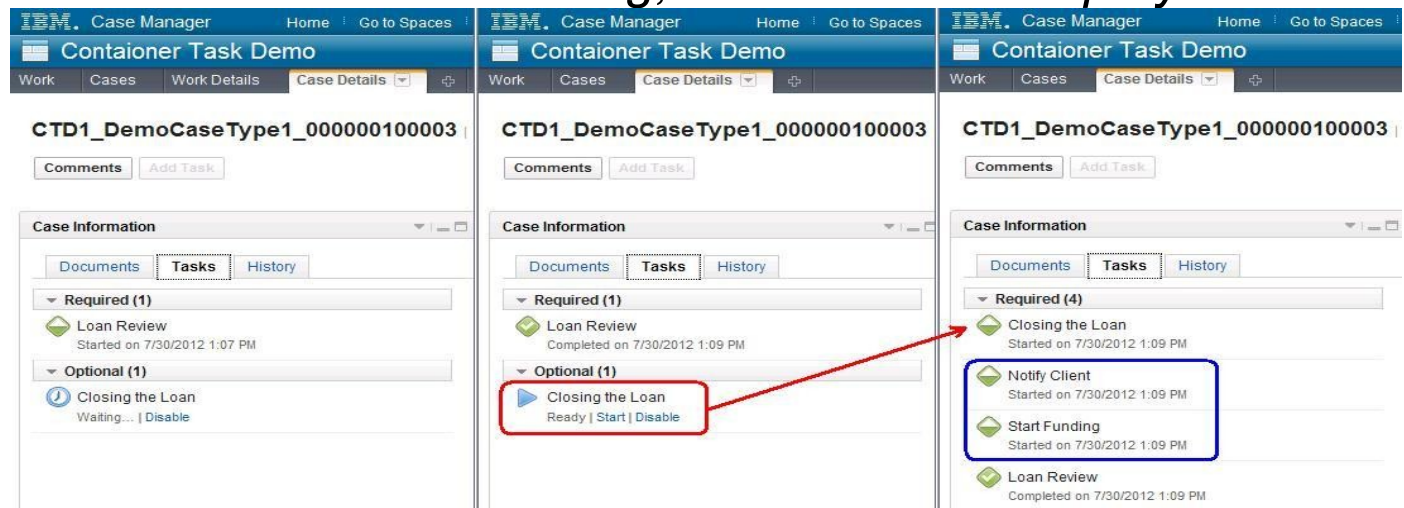


Subtasks Instantiation

- *When a container task moves to the working state, the subtasks are created and then put in the waiting state*
 - *Subtasks with no precondition or property condition is met successfully will become ready*
 - *Automatic tasks are moved to working*
 - *Subtasks with property update or document filing conditions will wait for the appropriate event to happen before being evaluated*
- *Using a discretionary container task is a powerful way to bring in a collection of predefined subtasks as needed for a case*
 - *As sometimes preconditions can not be always prescribed up front*
 - *Provides support for highly dynamic and adaptive case management based on case workers ad-hoc judgement and decisions on a case-by-case basis*

Case Client UI for Container Task and Subtask

- *No difference in regular task, container task, or subtask*
 - *Still a list of tasks in Tasks UI*
- *When a container task moves to working, the subtasks display in the Tasks list*



- *When creating a discretionary container task, only a new task name and description are required*
 - *Container Tasks do not have any associated task processes, just a list of subtasks*

Task Completion Semantics

- *A container task is complete only when*
 - *All required subtasks are completed or manually disabled (for manual tasks only)*
 - *Any working/running subtasks, even though optional, must be complete or stopped/aborted*
- *A case can become complete once*
 - *All required container tasks and tasks are completed or manually disabled (for manual tasks only)*
 - *Any working/running container tasks and top-level tasks, even though optional, must be complete or stopped/aborted*

Task Completion Semantics (cont.)

- *A waiting or ready state container task, even though it might have a required subtask, does not participate in the completion computation*
 - *As the subtask instance does not even exist until the container is in the working state*
 - *As needed, mark the container task as required*

Course Roadmap

- Task Type and Preconditions
- Container Task & Subtask
- ➔ Hidden Task
- Task Stop & Restart
- Course Summary

Background – Hidden Task

- *Use cases*
 - *Discretionary task types to be created by API only, and not manually by case workers*
 - *Not all task types involve case workers, therefore they do not need to be seen by the case workers*
- *All task types, including container task types, can be marked as hidden*
 - *Case Client will not show task types (when adding tasks) and task instances (in the Tasks list) if the task type is designated as hidden*
 - *Providing better Case Client usability and reducing case worker confusion*

Case Builder UI for Hidden Task

Discretionary tasks

Subtasks: 0

XXXX

Set: <None>

General Design Comment

* Name:
xxxx

* Unique Identifier
GA_ xxxx

Description:

This task starts:
☐ Automatically ☐ Manually ☒ Discretionally

This task is:
☒ Hidden
☐ Required

Assign to set:
<None>

Course Roadmap

- Task Type and Preconditions
- Container Task & Subtask
- Hidden Task
- ➔ Task Stop & Restart
- Course Summary

Background – Task Stop and Restart

- *Customer and partner requests for improving task management*
 - *Stop a running task*
 - *Restart a stopped task or failed task*
 - *Redo a task that completed successfully*
 - *Retry a task that failed to launch*

Task Stop & Restart Support

- *APIs are available to support customer/partner requests and business scenarios*
 - *Simplifies tasks that were manual and tedious in prior releases; e.g. following the procedure documented in a tech note for relaunching failed tasks*
- *Allows custom applications to support roles stop and restart processes associated with a case task*
 - *No ICM Case Client UI command buttons built into the out-of-the-box pages (i.e. not available unless customized and exposed as a functionality)*

Task Stop & Restart Scenarios

- *If a task process no longer needs to be completed, issue a Task Stop request to abandon the work items and any running processes gracefully*
- *If a task process has failed and can not be recovered, the case manager can issue a Task Stop request to clean up dangling work items and terminate the running processes properly*
 - *If needed, further issue a Task Restart to relaunch the tasks and start from the beginning of the process again*
- *If a task process is not working as desired, issue a Task Stop and then use the Restart to redo the process from the beginning*

Task Stop & Restart Scenarios (cont.)

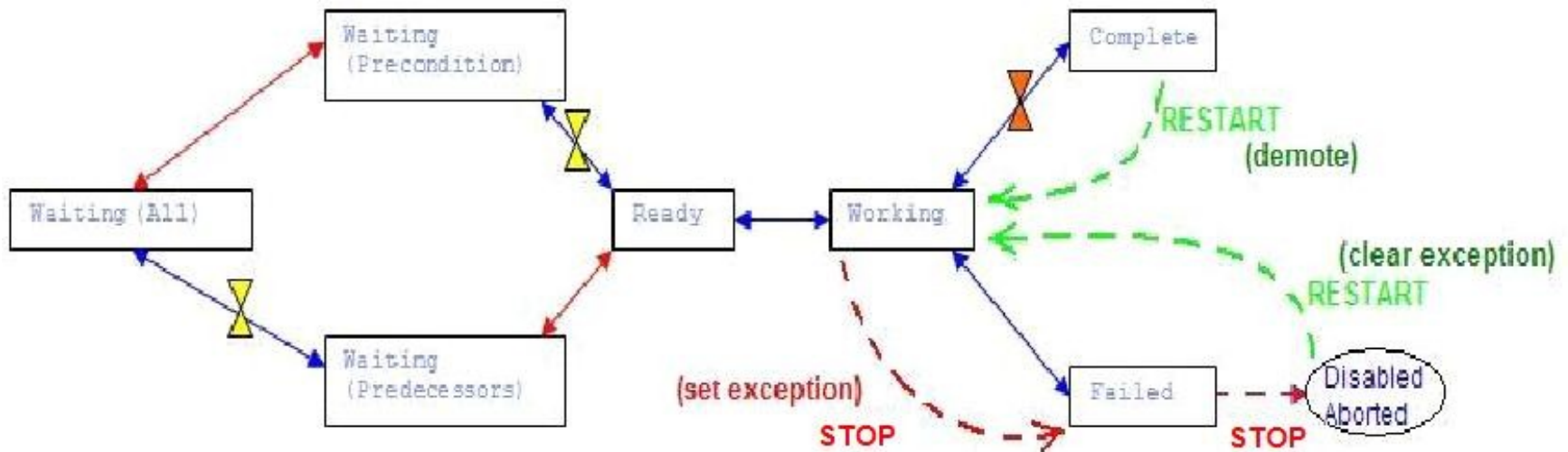
- *If a task completed successfully, issue a Task Restart and redo the whole process from the beginning again*
- *Use Task Restart to retry launching a task that originally failed to launch*
e.g. task moves to the working state and tries to launch the underlining workflow, the CPE server has an issue and fails to launch the process, as a result the task goes into a failed state

Task State

- *A PE process fails to launch*
 - *TaskState – Failed*
 - *LastFailureReason – “CmAcmError Launch Failed”*
- *PE process finishes successfully*
 - *TaskState – Completed*
- *PE sets an exception on the task instance when*
 - *A PE process fails, TaskState – Failed*
 - *A running PE process is stopped by a Task Stop request or the PE process already failed and the Task Stop is used to clean up the work items and terminate the PE process*
 - *task instance’s TaskState – Failed*
 - *task instance’s CmAcmDisabledState – Disabled Aborted (3)*

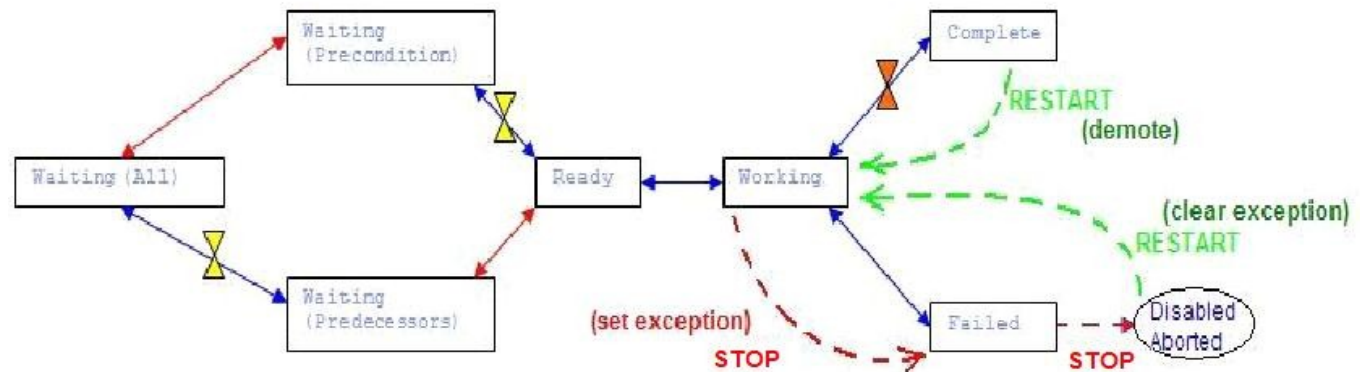
Task State Diagram

- CE supports
 - Completed task can be demoted back to Working state
 - Failed task with exception can be cleared and returned to a Working state



ICM API Supports

- Task Stop moves the running ICM Task from Working to Failed gracefully, or properly cleans up an already Failed Task
 - ICM Task ends up in the Failed state with Disabled Aborted (3)



- Task Restart
 - Demotes a successfully finished ICM Task from Complete back to the Working state
 - Allows (Task State in Failed AND Disabled Aborted) or (Task State in Failed with the well-known LastFailureReason string "CmAcmeError Launch Failed") to be put back to the Working state via Clear Exception

Task Stop and Restart Constraints

- *ICM only supports Stop or Restart invocation against a PE process-based regular task*
 - *No IBM BPM process support*
 - *Can not be called against a container task*
 - Regular subtasks can still be individually stopped and restarted*
 - *Discretionary tasks can only be stopped*
 - Create a new discretionary task to redo/repeat instead*
 - *Only stops or restarts the main PE workflow process associated with the Case Task, if the main process already creates other processes or has other side effects (for example, calls to external Web Services or uses a DBExecute, etc.) those will be left alone*
 - This is not a rollback mechanism*

Task Stop

- *Task Stop checks that the task is*
 - *Not a container task*
 - *Not an IBM BPM process task**and that CmAcmProcessInstanceId exists*
- *And that the task meets one of the following conditions*
 - *A PE task process is still running (TaskState – Working)*
 - *An already running PE process has failed (TaskState – Failed)*
- *If stopping the task is valid, the stop process invokes the PE API to terminate the PE process (by deleting work items and terminating the process gracefully)*
- *The Task ends up in the Failed and Disabled Aborted state*

Task Restart

- *Task Restart checks for*
 - *Not a container/compound task*
 - *Not an IBM BPM process task*
 - *Not a discretionary task*
 - *And one of the following conditions applies*
 - *A task PE process completed successfully*
 - *TaskState – Completed*
 - *A PE process failed to launch*
 - *TaskState – Failed*
 - *LastFailureReason – “CmAcmlError Launch Failed”*
 - *A running PE process has stopped, or failed then stopped*
 - *TaskState – Failed*
 - *CmAcmlDisabledState – Disabled Aborted (3)*

Task Restart (cont.)

- *If a Restart is valid, attempts to “Demote” or “Clear Exception” to put the Task back to the Working state*
 - *ICM/CE asynch event handler will then try to launch a new PE process as needed*
- Note: the current deployed/redeployed (i.e. most recently transferred) process definition is always used*
- *After a successful Restart, the ICM Task will have*
 - *TaskState – Working*
 - *CmAcmProcessInstanceId – new process id*
 - *CmAcmDisabledState – Enabled (0)*
 - *DateStarted – <original 1st start date time>*
 - *CmAcmLastDateRestarted -- <current restarted date time>*
 - *CmAcmRestartCount -- <restart count>*
 - *LastFailureReason, DateCompleted – null, not set*

Case Task and Case History

- *Case Task now tracks corresponding PE process id (CmAcmProcessInstanceId) when a task process is successfully launched*
- *Also, when the task is restarted the last restart date/time (CmAcmLastRestartDate), and the number of restarts (CmAcmRestartCount) are tracked on the task instance*
- *The restarts are also audited and captured in the CE audit log as Task Change State and Update events*
- *Case REST API has been updated to return additional information in the Tasks and Case History JSON payload*
- *Case Client Tasks List and History UI also understand additional information related to Tasks and Task State Changes*

Case Analytics

- *Once a task instance reaches “Complete” or “Failed” AND “Disabled Aborted” state, no additional task data is collected by Case Analytics*
If the Task restarts, the Case Analytics statistics data and report will be incorrect since it does not take into account the newly launched process and changed task states
- *This limitation is documented as a Known Issue*

Security Consideration

- *For a stop request to succeed, the user who makes the call must have read rights on all queues in the solution. By default, all users have read rights to queues, but if you have customized Process Engine security, you must add read rights for users who might call this API.*
- *For a restart request to succeed, the user who makes the call must have create rights for the roster of the solution.*

Split Case Feature - Introduction

- Purpose: allow case worker to split to a new case from an existing case
 - Be able to copy information associated with existing case
 - Similar functionality exists in Business Process Framework
- When the case is split, the new case
 - Can use the same or a different case type
 - Behaves as if added through Add Case functionality
- After the split, the new case & original case continue independently
 - Internally, a special relationship is established between the two cases
- Implemented via the new Split Case page in the Case Pages space

Split Case Functional Description


- How does one split a case?
 - Open an existing case in the Case Details page.
 - Click on Split Case button to start the split case operation.
- Split Case operation initiates from Case Details page
 - Split Case button: a button in Case Toolbar widget.
 - Similar to Add Case button: Can be split from same case type or other case types within the same solution.
 - Split Case button can be configured by edit Case Toolbar settings.

Case Details Page Screen shot

IBM Case Manager

Work

Cases

 Case AWM_AutoWarrantyClaim_000000100001

AWM_AutoWarrantyClaim_000000100001

Modified: 6/2/2014, 5:02 PM | AutoWarrantyClaim

Comments

Add Task

Add Custom Task

Split Case

Documents

Tasks

History

Add

Open

Actions

Home

No items to display

Customer Information

CustomerID

IX-99999

Customer Name

Bob Smith

Customer Address

9876 Havlestam Court

State

Texas

City

Dallas

Vehicle

VIN

JRX09876AAN

Vehicle Make

Toyota

Split Case Functional (cont.)

- Split Case Page - invoked when user clicks on the Split Case button in the Case Details page.
 - Created and registered when solution is deployed.
 - Contains 4 widgets: Case Toolbar, Split Case Data, Select Case Documents, and Original Case Data.
 - Split Case Data widget: show the case data properties that can be edited. Matching properties are pre-filled with values from the original case.
 - Original Case Data widget: Displays read-only property values from the original case.
 - Select Case Documents widget: Allows user to select documents from original case.

Go back one page
Right-click or pull down to show history

Work
Cases
Case AWM_AutoWarrantyClaim_000000100001
Split Case

Split From: AWM_AutoWarrantyClaim_000000100001 | AutoWarrantyClaim

Split Case:

Customer Information
CustomerID
IX-99999
Customer Name
Bob Smith
Customer Address
9876 Havlestam Court
State
Texas
City
Dallas

Vehicle

Original Case:

Customer Information
CustomerID
IX-99999
Customer Name
Bob Smith
Customer Address

Select the documents that you want to add to the new case. The documents are not removed from the original case. Documents to be added:


Home

	Name
No items to display	

Split Case Page Selction screen shot (Case Builder)

Go back one page
Right-click or pull down to show history

Manage Solutions \ Auto Warranty Management



Auto Warranty Management

Warranty Management System for Automotive manufacturer
Solution prefix: AWM
Created by P8Admin
Created on January 15, 2014

Properties
Roles
In-baskets
Document Types
Pages
Case Types

?
Add Page
OK All

▶ Solution Pages

▶ Case Details Pages

▶ Add Case Pages

▼ Split Case Pages

Page Name ^
Description ^

Split Case
Create a case that reuses property values from an existing case and edit data for

▶ Add Task Pages

▶ Work Details Pages

▶ Custom Task Pages

Split Case Page Designer screen shot (Case Builder)

Go back one page
Right-click or pull down to show history

Split Case

▼ Case Widgets

- Attachments
- Case Information
- Case List
- Case Toolbar
- Content List
- Form
- In-baskets
- Instruction
- Original Case
- Process History

▼ Utility Widgets

- Script Adapter
- Website Viewer

Case Toolbar

Height: [Automatic](#)
[0 Incoming Events](#)
[0 Outgoing Events](#)

Split Case

Height: [400Pixels](#)
[0 Incoming Events](#)
[0 Outgoing Events](#)

Select Case Documents

Height: [100%](#)
[0 Incoming Events](#)
[0 Outgoing Events](#)

Original Case

Height: [215Pixels](#)
[0 Incoming Events](#)
[0 Outgoing Events](#)

Post-Split Case Behavior

- Both split case and original case continue independently:
 - Both cases must be completed separately.
 - Tasks from the original case continue to run, new tasks for the split case are initiated in the normal fashion, as if you have added case.
- Selected documents are filed into new split case at the case root level.
- Internally, a relationship is established between the new split case & the original case. This relationship can be viewed from the history tab of the Case Info widget in the Case Details page of the split case or the original case.