IBM Case Manager 5.2

External Data Integration

Content

- Course Overview
  - Overview of the external data feature
  - How ICM integrates with external data
  - External data service and supported features
  - Registering an external data service
  - Sample external data services

Course Roadmap

- ➔ Overview of the external data feature
- – How ICM integrates with external data
- – External data service and supported features
- – Registering an external data service
- – Sample external data services
- – Course summary

3

Overview of External Data Integration

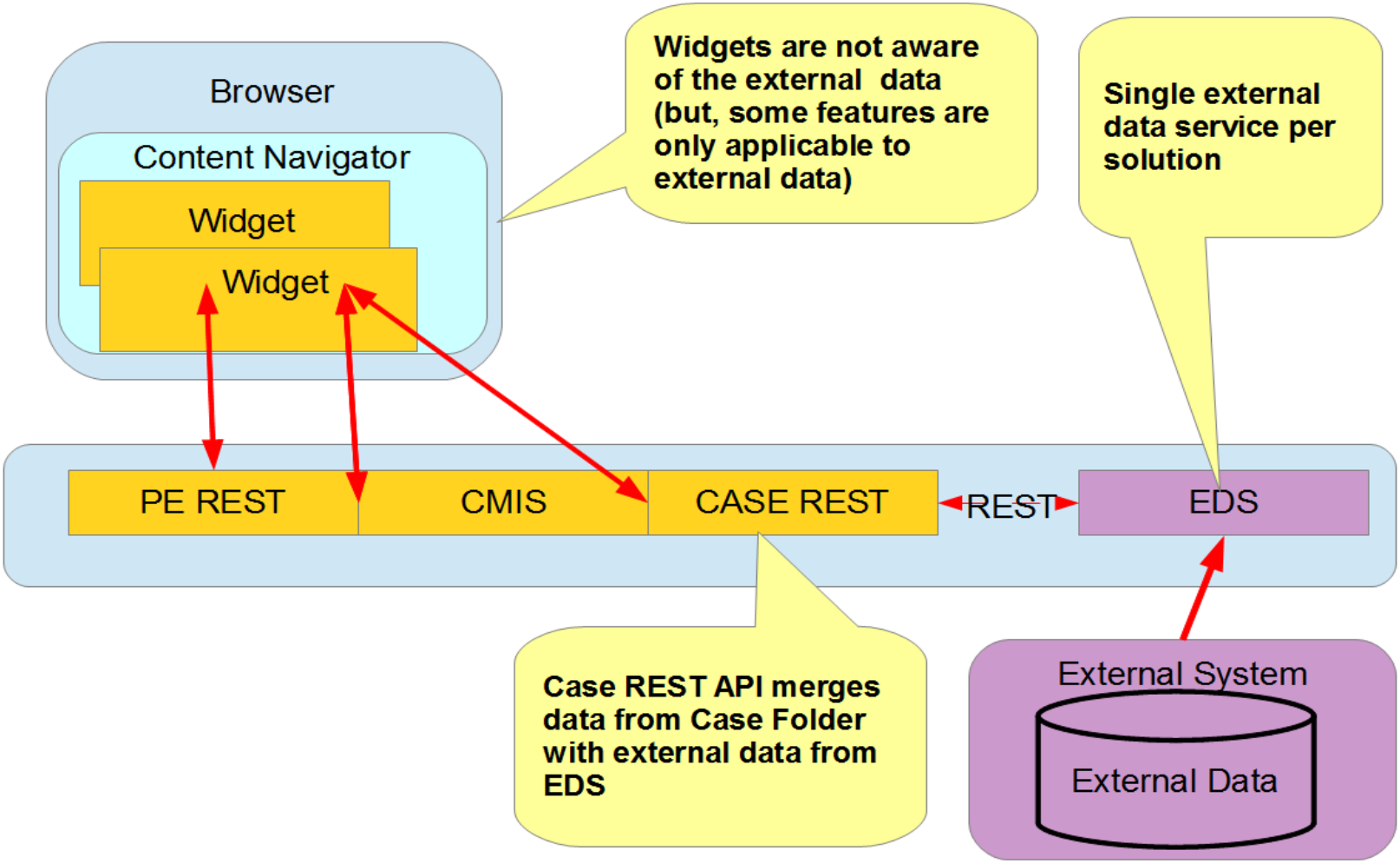
- ICM communicates with an External Data Service (EDS), but the customer must provide the particular implementation of the EDS.
- Allows certain information about case properties to come from an external data source
  - Choice lists can be generated from data stored externally
  - Attributes such as minimum/maximum values can be overridden
- The values of properties can be influenced by the EDS, but the resulting values are still stored in CE
- EDS can manage dependencies between properties
  - Attributes such as choice lists can depend on values of other properties

Course Roadmap

- Overview of the external data feature
- ➔ How ICM integrates with external data
- External data service and supported features
- Registering an external data service
- Sample external data services
- Course summary

5

Integration Into ICM



Integration Into ICM

- Only properties on cases are integrated with external data
- A solution can be associated with one EDS
  - EDS can manage some or all of the case types in the solution.
- Integration with the external data service is in the middle tier (Case REST)
- Widgets are unaware if some of the information about case properties comes from external data
  - Some enhancements in the widgets are only applicable to features that use external data
    - e.g. defining dependencies between properties

Case REST Support

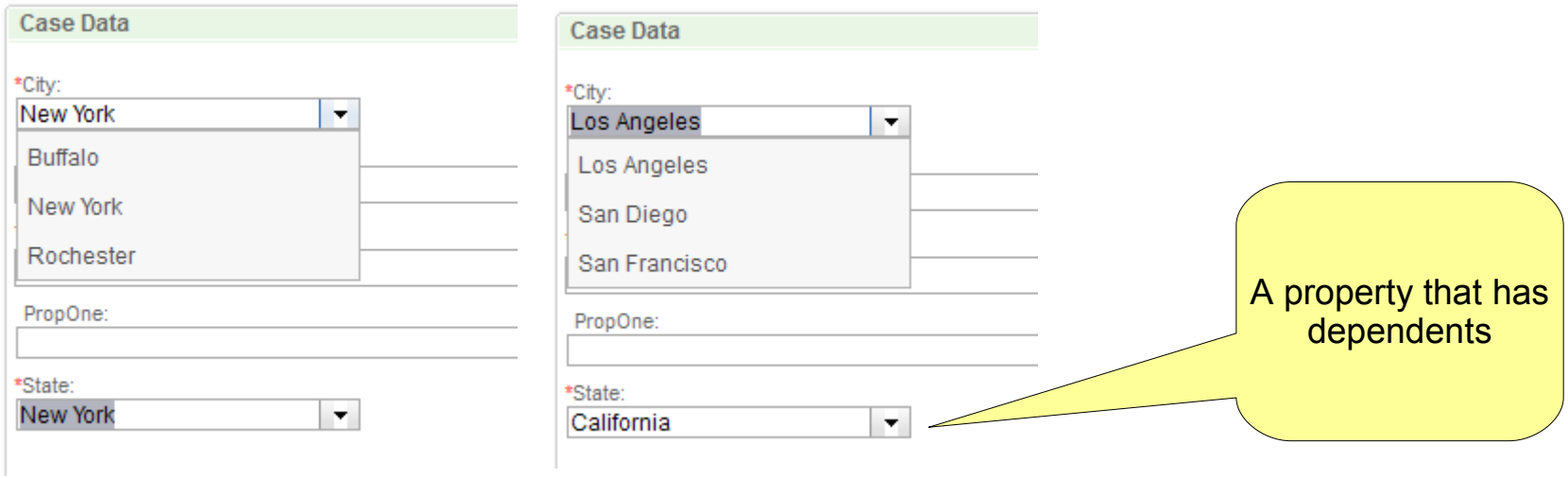
- Case REST merges information from the external data source and then returns the merged information in the payloads to the client

```
{
      "SymbolicName": "DH2_State",
      "DisplayName": "State",
      "Value": null,
      "PropertyType": "string",
      "Cardinality": "single",
      "Required": true,
      "Hidden": false,
      "MaxLength": 2,
      "ChoiceList": {
            "DisplayName": "StateChoiceList",
            "Choices": [
                  {
                        "ChoiceName": "New York",
                        "Value": "NY"
                  },
                  {
                        "ChoiceName": "California",
                        "Value": "CA"
                  },
                  ...
            ]
      },
      "HasDependentProperties": true
},
```

Certain attributes can come from an external data service

Widgets Support

- Some enhancements in the widgets to deal with new features applicable to external data



Course Roadmap

- Overview of the external data feature
- How ICM integrates with external data
- ➔ External data service and supported features
- Registering an external data service
- Sample external data services
- Course summary

10

External Data Service

- Implements a RESTful interface communicating over the HTTP protocol
- A POST method
  - Input payload has current working values and request mode
    - Working values may be persisted values or values user has entered so far
    - Request mode indicates for example whether properties are being fetched initially or dependent changes are being fetched
  - Response payload contains overridden working values or property attributes

```
POST
http://myicmserver.com/testservice/ICMEDREST/type/D
H2_MyCase
{
      "repositoryId": "CMTOSDH",
      "requestMode": "initialNewObject",
      "properties": [
            ...
            {
                  "symbolicName": "DH2_State",
                  "value": null
            },
            {
                  "symbolicName": "DH2_PropOne",
                  "value": null
            },
            {
                  "symbolicName": "DH2_City",
                  "value": null
            }
      ]
}
```

Request Content

Response

```
{
    "externalDataIdentifier": "-1,0",
    "properties": [
        {
            "symbolicName": "DH2_State",
            "required": true,
            "maxLength": 2,
            "hasDependentProperties": true,
            "value":"NY",
            "choiceList": {
                "displayName": "StateChoiceList",
                "choices": [
                    {
                        "displayName": "New York",
                        "value": "NY"
                    },
                    {
                        "displayName": "California",
                        "value": "CA"
                    },
                    ...
```

Assigning Property Attributes from External Data

- Properties are whatever is defined in the target object store
  - Makes no difference whether property was originally defined in Case Builder or whether it was a reuse property
- EDS can specify a choice list for a property
  - Cannot override the choice list if one is assigned in CE
- Can override minimum and maximum value and maximum length attributes
  - Can only make it more restrictive than the underlying CE value (e.g. minimum value greater than underlying value)
- Can specify a required attribute of true if the underlying setting is false
  - Cannot make a required property non-required
- Can override the hidden attribute
  - e.g. show or hide properties based on other input values

Influencing Property Values

- EDS can influence what value is saved for a property by modifying the working value
  - Specify a working value different than the default for a new case, the current value of an existing case, or the value a user previously entered
  - For example EDS may change the value if the current working value is invalid based on other attributes that may have changed
  - The final value is still persisted in CE
- Can specify a display mode of read-only, ensuring it is the value persisted
- Can specify custom validation information
  - For example AccountNumber does not represent a valid customer. EDS supplies custom validation error rather than assigning a new working value.

Dependencies Between Properties

- A property can be designated with a “HasDependentProperties” attribute of true
- Whenever the working value changes, a round trip is made by the client to fetch dependent changes to other properties
  - Example: State and City properties. When value of state changes a different choice list is assigned to City.
  - Example: available plans based on account balance. A balance above some threshold may effect the new plans a customer can choose from.
- EDS responds with any properties that have changes based on what the client is currently holding
  - Different working values
    - Current working values passed in the input payload. EDS changes if appropriate.
  - Different attributes such as choice list, minimum/maximum, etc.
    - EDS uses the External Data Identifier to determine when attributes have changed from what was returned previously
  - Custom validation errors

Case REST For Fetching Dependent Changes

POST to the particular case type resource. Input payload contains the current working values and external data identifier

```
POST /CaseManager/CASEREST/v1/casetype/DH2_MyCase
{
    "TargetObjectStore": "CMTOSDH",
    "ExternalDataIdentifier": "-1,0",
    "Properties": [
        ...
        {
            "SymbolicName": "DH2_State",
            "Value": "CA"
        },
        {
            "SymbolicName": "DH2_PropOne",
            "Value": null
        },
        {
            "SymbolicName": "DH2_City",
            "Value": "New York"
        },
    ]
}
```

e.g. State property was modified. Client submits all values. EDS determines appropriate changes based on previous external data identifier.

A corresponding call is made by the middle tier to the EDS

Response includes the properties that have changed and a new external data identifier

```
{
    "Properties": [
        {
            "SymbolicName": "DH2_City",
            "DisplayName": "City",
            "Value": "Los Angeles",
            "DisplayMode": "readwrite",
            "PropertyType": "string",
            "Cardinality": "single",
            "Updatability": "readwrite",
            "Required": true,
            "Hidden": false,
            "MaxLength": 64,
            "ChoiceList": {
                "DisplayName": "CityChoiceList",
                "Choices": [
                    {
                        "ChoiceName": "Los Angeles",
                        "Value": "Los Angeles"
                    },
                    {
                        "ChoiceName": "San Diego",
                        "Value": "San Diego"
                    },
                    ...
                ]
            },
            "HasDependentProperties": false
        }
    ],
    "ExternalDataIdentifier": "1,0"
}
```

External Data Identifier

- A special attribute returned in the payload is ExternalDataIdentifier
- This is opaque state meaningful to the EDS
- EDS should capture any state that indicates what configurations of properties it returns – assigned choice lists, etc.
  - Allows changes to be determined when returning dependent changes
  - Applicable when the attributes can differ based on current working values
  - An example implementation (used by sample EDS):
    - EDS has conditions that select property configurations based on working values (e.g. when State = “CA” use this configuration; when State = “NY” ...)
    - Capture the indices of the matching conditions in the opaque identifier
    - Based on matching indices returned previously, EDS knows what is different based on new working values

External Data Identifier

- EDS is free to capture any other state it wants maintained in the opaque identifier
- Requires client to maintain the value returned by the EDS (passed through by the Case REST)
  - EDS is otherwise stateless
  - Passes the last value returned by the EDS when making additional calls.
    - To fetch dependent changes
    - When making the final call to create or save the case

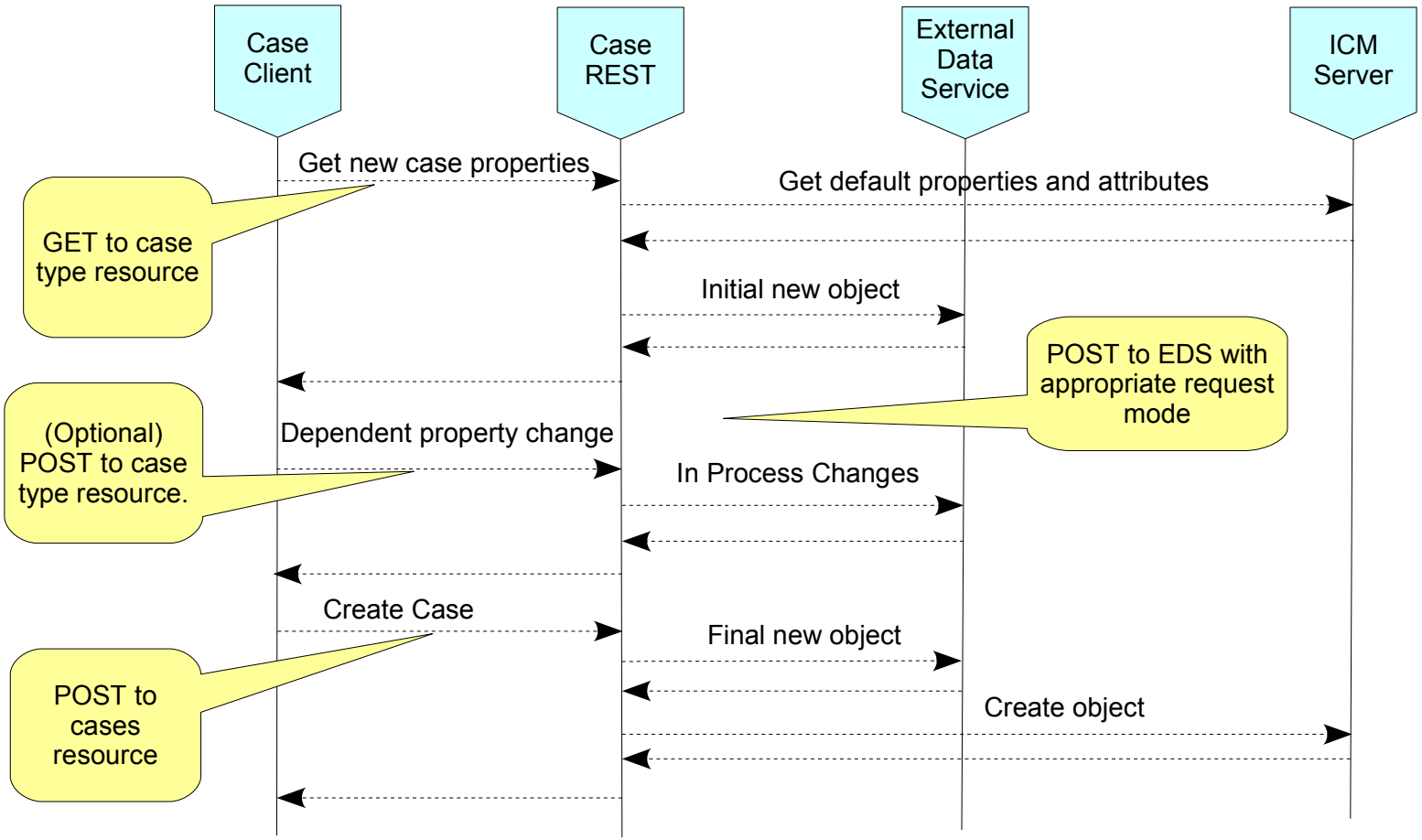
Client Context

- Client can pass an optional ClientContext portion.
- ClientContext is transparent to Case REST and is passed through to the EDS
- EDS can have a relationship with a specific client and understand particular context keys
- Case Widgets generate a client context when editing work item fields associated with case properties.
  - Context indicates workflow step information. EDS can specify property information based on this context if it chooses.

```
"ClientContext":
{
    "connectionPoint" : "ConnPt1",
    "stepId": 1,
    "mapName": "Workflow",
    "workflowNumber": "51C42F6F286CD246A811E633EE30A13B",
    "workflowName": "EX3_task1",
    "caseTaskId": "{85EB0EB4-47E3-410B-9F31-5E053ABB399C}",
    "stepName": "Step",
    "workObjectNumber": "51C42F6F286CD246A811E633EE30A13B",
    "authoredMapName": "Workflow",
    "queueName": "EX3_role1",
    "originator": "administrator",
    "subject": "t5",
    "launchDate": "2011-04-28T04:16:22Z",
    "role":"role1"
}
```

A collection of key/value pairs

Example Interaction -- Case Creation



Authentication

- EDS can authenticate against the logged on user using the same single sign-on authentication as the other Case Manager components – e.g. Case Manager Client, Case Manager REST
- Middle-tier propagates the LtpaToken2 cookie in the request to the EDS
- If EDS is not in the same profile as the other Case Manager components, import the LTPA keys
  - From WAS Console for profile running CE server, export the LTPA keys (Global security / LTPA)
  - From WAS Console for profile running EDS, import the LTPA keys

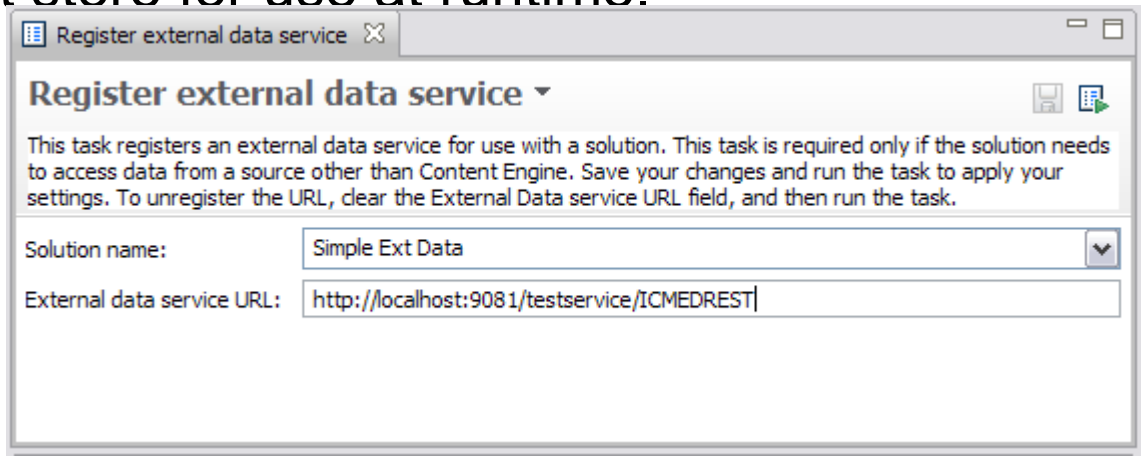
Course Roadmap

- Overview of the external data feature
- How ICM integrates with external data
- External data service and supported features
- ➔ Registering an external data service
- Sample external data services
- Course summary

21

Registering an External Data Service

- A particular EDS is installed using a procedure unique to that EDS
  - Probably involves installing a Web Application in a WebSphere Application Server profile
- Use Case Manager Configuration Tool to register the URI for a solution
- Deploy the solution. The URI is captured in the target object store for use at runtime.



Course Roadmap

- Overview of the external data feature
- How ICM integrates with external data
- External data service and supported features
- Registering an external data service
- ➔ Sample external data services
- Course summary

23

Sample External Data Services

- At least two sample External Data Services are available on developerWorks
- Source code is included
- To find the samples:
  - Go to http://www.ibm.com/developerWorks
  - Search on: IBM Case Manager external d

Property configurations can be defined that specify or override various property attributes

Sample External Data Service

- We now look at the first sample (by Dave Hanson)
- Java Servlet-based web application
- Uses an XML configuration file to control what case types and properties are managed

```
<Property identifier="St">
    <SymbolicName>SOL_State</SymbolicName>
    <PropertyType>string</PropertyType>
    <Cardinality>single</Cardinality>
    <MaximumLength>2</MaximumLength>
    <Required>true</Required>
    <HasDependentProperties>true</HasDependentProperties>
    <ChoiceList>
        <DisplayName>StateChoiceList</DisplayName>
        <Choices>
            <Choice>
                <Name>New York</Name>
                <Value>NY</Value>
            </Choice>
            <Choice>
                <Name>California</Name>
                <Value>CA</Value>
            </Choice>
            ...
        </Choices>
    </ChoiceList>
</Property>
```

Property configurations can be defined that specify or override various property attributes

Sample EDS – Dependent Properties

```
<ConditionalPropertyName>SOL_State</ConditionalPropertyName>
<ConditionalPropertySets>
    <ConditionalPropertySet>
        <Equals>NY</Equals>
        <PropertySet>
            <StaticProperties>
                <PropertyRef identifier="Ct@St=NY" />
            </StaticProperties>
        </PropertySet>
    </ConditionalPropertySet>
    <ConditionalPropertySet>
        <Equals>CA</Equals>
        <PropertySet>
            <StaticProperties>
                <PropertyRef identifier="Ct@St=CA" />
            </StaticProperties>
        </PropertySet>
    </ConditionalPropertySet>
    ...
</ConditionalPropertySets>
<DefaultPropertySet>
    <StaticProperties>
        <Property>
            <SymbolicName>SOL_City</SymbolicName>
            <PropertyType>string</PropertyType>
            ...
            <Hidden>true</Hidden>
        </Property>
    </StaticProperties>
</DefaultPropertySet>
```

```
<Property identifier="Ct@St=NY">
    <SymbolicName>SOL_City</SymbolicName>
    <PropertyType>string</PropertyType>
    ...
    <ChoiceList>
        <DisplayName>CityChoiceList</DisplayName>
        <Choices>
            <Choice>
                <Name>Buffalo</Name>
                <Value>Buffalo</Value>
            </Choice>
            <Choice>
                <Name>New York</Name>
                <Value>New York</Value>
            </Choice>
            ...
```

```
<Property identifier="Ct@St=CA">
    <SymbolicName>SOL_City</SymbolicName>
    <PropertyType>string</PropertyType>
    ...
    <ChoiceList>
        <DisplayName>CityChoiceList</DisplayName>
        <Choices>
            <Choice>
                <Name>Los Angeles</Name>
                <Value>Los Angeles</Value>
            </Choice>
            <Choice>
                <Name>San Diego</Name>
                <Value>San Diego</Value>
            </Choice>
            ...
```

Sample EDS – Value Handling

```
<RenderedReadOnlyValue>1000</RenderedReadOnlyValue>
```

Always ensure a specific value with DisplayMode readonly

```
<ValueIfNew>Los Angeles</ValueIfNew>
```

Override initial value for a new case

```
<ValueIfInvalid handling="replaceValue">Los Angeles</ValueIfInvalid>
```

Value to specify if current working value is invalid

```
<ValueIfInvalid handling="returnMessage"/>
```

"Pseudo" custom validation. Returns a custom validation error if current value is invalid. However validation is based on normal criteria – min/max, choice list, etc.

Course Roadmap

- Overview of the external data feature
- How ICM integrates with external data
- External data service and supported features
- Registering an external data service
- Sample external data services
- ➔ Course summary

28

Summary

- Using external data integration, information about case properties can come from an external data source
- A customer provides an implementation of an external data service (EDS)
- Integration with the EDS is in the middle-tier (Case REST)
- One EDS can be registered for a solution
- Sample EDS's are provided on developerWorks