

---

# Taking the Logical to its Physical Conclusion

---

**Sydney Osborne**  
**Robert Lutolf**

California Department of Health Services

Sept 2006

# Presenters



- **Sydney Osborne**

Software Engineer for the Department of Health Services (CA-DHS) Childhood Lead Poisoning Prevention Branch (CLPPB) Response and Surveillance System for Childhood Lead Exposure (RASSCLE II) project.

- **Bob Lutolf**

President, Gensa Corporation.  
Application Development Manager  
for RASSCLE II. Designed  
RASSCLE II Physical Database.



# Presentation Background

---

- The Response and Surveillance System for Childhood Lead Exposure (RASSCLE) is used by Childhood Lead Poisoning Prevention Branch (CLPPB) to collect information on children found to have elevated blood lead levels.
- Our legacy RASSCLE system is in the final stages of being re-engineered as a state-wide, web-based information system (RASSCLE II).
- We chose the PHIN Logical Data Model (PHLDM) to guide our design of the RASSCLE II database.
- Last year, we presented an overview of the process used to derive the RASSCLE II Logical Data Model from the PHIN Logical Data model with the following aims:

# RIIDDM Design Constraints

---

1. Meet the documented RASSCLE II information and functional requirements.
2. Remain consistent with the PHIN Logical Data Model and associated data standards.
3. Use a modeling style that anticipates changes in RASSCLE II information and functional requirements.
4. Position the database to be useful for other purposes within the California Department of Health Services.
5. Strike a balance between extensibility, programming complexity, and system performance.

# Presentation Overview

---

In this presentation, we will:

- Describe why and how we used the PHLDM/RIILDDM Observation tables largely unchanged in the Physical Database Model
- Show the approach we took at the SQL code level to work with these data.
- *Warning: The second half of this talk will be code-heavy (SQL, XML and C# code).*

# PHLDM Primer

---

- Entities (e.g., Person, Place)
- Acts (e.g., Home Visit, Send Alert)
  - ObservationEvents
  - ActDefinitions
  - CommunicationEvents
  - InterventionEvents
  - Etc.
- Act Participations (Register for Alert)
- Roles (Provider, Patient)
- Relationships (Mother, Brother)

# RASSCLEII Business Requirements

---

- Blood Test Results
- Data on Lead Poisoned Children
  - Insurance Coverage/Program enrollment
  - General Health Information
  - Possible Sources of Exposure
  - Use of Common Lead-Containing things
    - Home Remedies, Ceramics, Cosmetics, Foods
  - Exposure to Household members who work with Lead
  - Construction and Remodeling history of home
  - Etc.

# RASSCLEII Logical Data Model

---

- Observation Events
  - Blood Observation Events
  - Clinical Observation Events
  - Construction Observation Events
  - Environmental Sample Observation Events
  - Other Environmental Observation Events

# Detour: Vocabulary and Coded Values

---

- Vocabulary Domain: Sex
  - codedValue:1000 = Male
  - codedValue:1001 = Female
  - codedValue:1002 = Undifferentiated
- Vocabulary Domain: Observation Events
  - codedValue: 9990 = Ceramic Exposure Event
  - codedValue: 9991 = Paint Sampling Event
  - codedValue: 9992 = Some Other Event
- Ceramic ObservationEvent.typeCode = 9990

# Observation Events

---

- Identifier
- One or more foreign keys (e.g., to Public Health Case table)
- ActivityFrom and ActivityTo fields
- Status Code
- Type Code
- Parent Observation Event Identifier (self-referencing foreign key for nested events)
- Sequence Number

# Observation Value

## A\_EnvironmentalObservationValue



identifier

environmentalObservationEventIdentifier

datatypeCode

derivationText

sequenceNumber

typeCode

uncertainValueProbabilityQuantity

valueCode

valueDateTime

valueImage

valueIndicator

valueInteger

valueReal

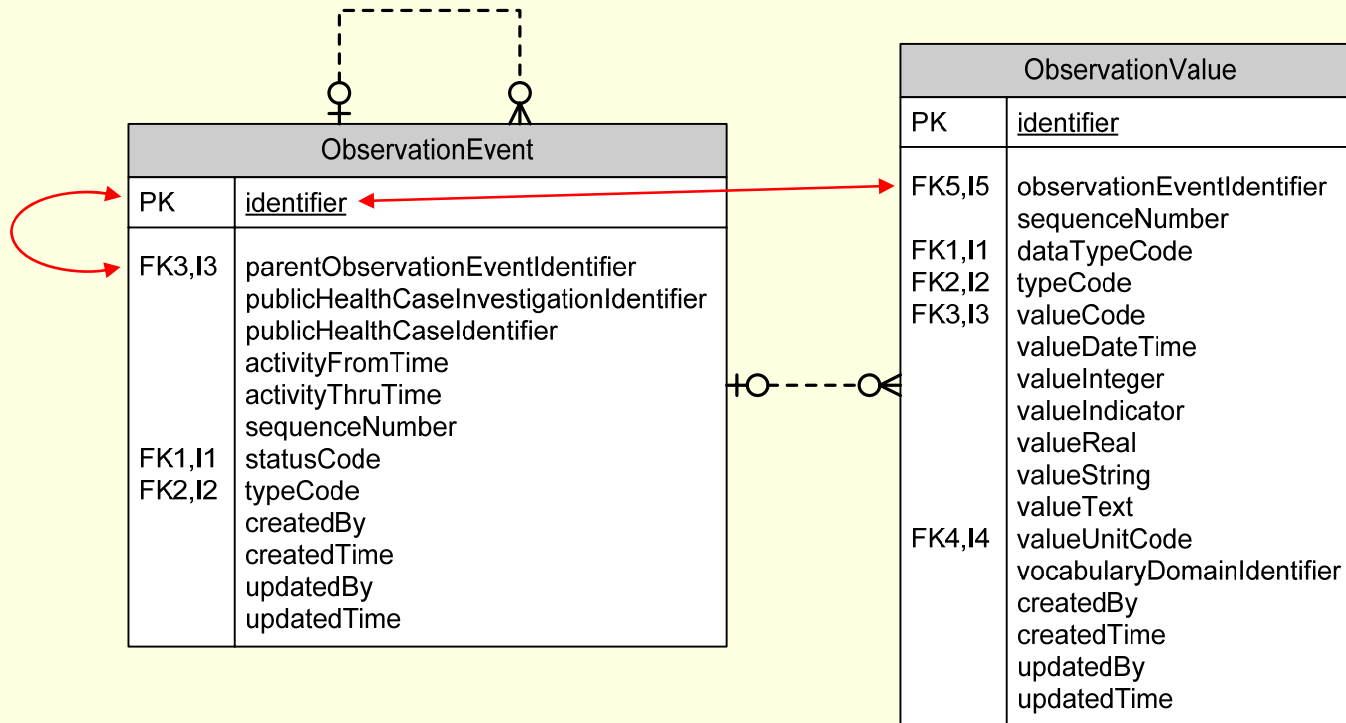
valueString

valueText

valueUnitCode

vocabularyDomainIdentifier

# Observation Events and Values



# Six Top-Level Observation Event records

Parent Observation Event	Event Identifier	Observation Event Type	Activity From Time	Activity Through Time
▶	{0D3A8DAC-82CC-44F8-9629-6E8B988EF3A3}	Pottery-Utensils Event		
	{41AEBB01-3879-4747-8DE7-6BDBB53F488A}	Food Drink		
	{93DA88E8-B711-4BE0-85F9-0393C281CE4E}	Housing Characteristics		
	{B60DCFC0-4AFF-441B-9C15-749163ECDA4E}	Pottery-Utensils Event		
	{BAF75B44-4EF9-485E-81DF-6D1ABE4E2EB1}	Hobbies		
	{EC489803-EAC0-4C3C-896B-755AC441D243}	Cosmetics		

Top level Observation Events have no Parent Observation Event

# These Events can have Child Values

Parent Observation Event	Event Identifier	Observation Event Type	Activity From Time	Activity Through Time			
	{0D3A8DAC-82CC-44F8-9629-6E8B988EF3A3}	Pottery-Utensils Event					
	{41AEBB01-3873-4747-8DE7-6BDBB53F488A}	Food Drink					
	{93DA88E8-B711-4BE0-85F9-0393C281CE4E}	Housing Characteristics					
Observation Event	Type	Data Type	Coded Value	String Value	Date Value	Integer Value	Real Value
{0D3A8DAC-82CC-44F8-9629-6E8B988EF3A3}	Usually Eats Out of	Character String		Plastic			
{0D3A8DAC-82CC-44F8-9629-6E8B988EF3A3}	Household Ceramic use	Coded Value	Yes				
	{E4489693-EAC0-4C3C-896B-753AC441D243}	Cosmetics					

# Events can also have child Events

Parent Observation Event	Event Identifier	Observation Event Type	Activity From Time	Activity Through Time
	{0D3A8DAC-82CC-44F8-9629-6E8B988EF3A3}	Pottery-Utensils Event		

Observation Event	Type	Data Type	Coded Value	String Value	Date Value	Integer Value	Real Value
{0D3A8DAC-82CC-44F8-9629-6E8B988EF3A3}	Usually Eats Out of	Character String		Plastic			
{0D3A8DAC-82CC-44F8-9629-6E8B988EF3A3}	Household Ceramic use	Coded Value	Yes				

Values:

Parent Observation Event	Seq No	Event Identifier	Observation Event Type	Activity From Time	Activity Through Time
{0D3A8DAC-82CC-44F8-9629-6E8B988EF3A3}	1	{3B5B733E-FA43-4083-AE30-A5D576CC8846}	Ceramic Use		
{0D3A8DAC-82CC-44F8-9629-6E8B988EF3A3}	2	{462FBE40-CA15-4818-A70B-2AF830E8753F}	Ceramic Use		
		{BAF75B44-4EF9-485E-81DF-6D1ABE4E2EB1}	Hobbies		
		{EC489803-EAC0-4C3C-896B-755AC441D243}	Cosmetics		

Child Events:

# Child Events can have Events and Values

Parent Observation Event	Event Identifier	Observation Event Type	Activity From Time	Activity Through Time
	{0D3A8DAC-82CC-44F8-9629-6E8B988EF3A3}	Pottery-Utensils Event		

Value Records:

Observation Event	Type	Data Type	Coded Value	String Value	Date Value	Integer Value	Real Value
{0D3A8DAC-82CC-44F8-9629-6E8B988EF3A3}	Usually Eats Out of	Character String		Plastic			
{0D3A8DAC-82CC-44F8-9629-6E8B988EF3A3}	Household Ceramic use	Coded Value	Yes				

Child Event:

Parent Observation Event	Seq No	Event Identifier	Observation Event Type	Activity From Time	Activity Through Time
{0D3A8DAC-82CC-44F8-9629-6E8B988EF3A3}	1	{3B5B733E-FA43-4083-AE30-A5D576CC8846}	Ceramic Use		

Value Records:

Observation Event	Type	Data Type	Coded Value	String Value	Date Value	Integer Value	Real Value
{3B5B733E-FA43-4083-AE30-A5D576CC8846}	Test Kit	Coded Value	Positive				
{3B5B733E-FA43-4083-AE30-A5D576CC8846}	Sample Collected	Coded Value	Yes				
{3B5B733E-FA43-4083-AE30-A5D576CC8846}	Ceramic use description	Character String		Drinks cocoa from the mug			
{3B5B733E-FA43-4083-AE30-A5D576CC8846}	Ceramic Item Name	Character String		Ceramic mug			

Child Event:

Parent Observation Event	Seq No	Event Identifier	Observation Event Type	Activity From Time	Activity Through Time
{0D3A8DAC-82CC-44F8-9629-6E8B988EF3A3}	2	{462FBE40-CA15-4818-A70B-2AF830E8753F}	Ceramic Use		

Value Records:

Observation Event	Type	Data Type	Coded Value	String Value	Date Value	Integer Value	Real Value
{462FBE40-CA15-4818-A70B-2AF830E8753F}	Test Kit	Coded Value	Negative				
{462FBE40-CA15-4818-A70B-2AF830E8753F}	Sample Collected	Coded Value	Yes				
{462FBE40-CA15-4818-A70B-2AF830E8753F}	Ceramic use description	Character String		Used for serving soups			
{462FBE40-CA15-4818-A70B-2AF830E8753F}	Ceramic Item Name	Character String		Ceramic Label			

	{41AEBB01-3879-4747-8DE7-6BDBB53F488A}	Food Drink			
	{93DA88E8-B711-4BE0-85F9-0393C281CE4E}	Housing Characteristics			
	{B60DCFC0-4AFF-441B-9C15-749163ECDA4E}	Pottery-Utensils Event			
	{BAF75B44-4EF9-485E-81DF-6D1ABE4E2EB1}	Hobbies			
	{EC489803-EAC0-4C3C-896B-755AC441D243}	Cosmetics			

# Ceramic Use

Do you have any imported or handmade ceramics in the household? (List those with which case comes into contact)

Yes \*

Out of what does the case usually eat and drink?

Plastic

#	Imported or handmade ceramic	Test Kit Result	Delete
<a href="#">2</a>	<b>Ceramic Ladle</b>	<b>Negative</b>	<a href="#">Delete</a>
<a href="#">1</a>	Ceramic mug	Positive	<a href="#">Delete</a>

Imported or handmade ceramics

Ceramic mug

Sample Collected?

Yes

Describe its use

Drinks cocoa from the mug

Test Kit result

Positive

# Ceramic Events and Values

---

Ceramics Event (Non-Repeating)

Family Has Imported Ceramics? (valueCode)

What does patient eat out of? (valueString)

Ceramic Item Event (Repeating)

Brief Description (valueString)

Was sample collected? (valueCode)

Description of ceramic use (valueString)

Test-Kit result (valueCode)

# Flexibility

---

## Ceramics Event (Non-Repeating)

Family Has Imported Ceramics? (valueCode)

What does patient eat out of? (valueString)

## Ceramic Item Event (Repeating)

Brief description of ceramic item (valueString)

Description of ceramic use (valueString)

Frequency of use (valueInt + valueUnitCode)

Country of origin (valueCode)

## Samples Collected Event (Repeating)

Sample Number (valueInteger)

Sample Result (valueReal + valueUnitCode)

# FOR XML and OpenXML

---

- XML vs. Recordsets
- XML supports multiple levels of nested data
- SQL Server functionality to return data as XML
- FOR XML clause returns recordsets as XML
- Complementary OpenXML enables working with XML as a virtual table

# Retrieving Observation Data as XML

```
CREATE PROCEDURE up_sxml_ClinicalObservation1
(
    @clinicalPublicHealthCaseIdentifier int = NULL,
    @event uniqueidentifier = NULL,
    @topLevelEventTypeCode int
)
AS
SELECT
    event.identifier, event.parentObservationEventIdentifier, event.sequenceNumber, event.typeCode,
    event.updatedTime, event.clinicalPublicHealthCaseIdentifier, event.updatedBy,

    value.identifier, value.clinicalObservationEventIdentifier, value.sequenceNumber, value.typeCode,
    value.valueCode, value.valueInteger, value.valueDateTime, value.valueString, value.valueReal,
    value.valueIndicator, value.valueUnitCode
FROM (
    SELECT identifier, clinicalPublicHealthCaseIdentifier, parentObservationEventIdentifier,
           sequenceNumber, typeCode, updatedTime, updatedBy
    FROM A_ClinicalPublicHealthCaseObservationEvent event
    WHERE typeCode = @topLevelEventTypeCode
           AND ((@event IS NULL) OR (identifier = @event))
           AND ((@clinicalPublicHealthCaseIdentifier IS NULL)
                OR (clinicalPublicHealthCaseIdentifier = @clinicalPublicHealthCaseIdentifier))
    ) as event
LEFT JOIN
(
    SELECT value.identifier, value.sequenceNumber, clinicalObservationEventIdentifier, value.typeCode,
           valueCode, valueInteger, valueDateTime, valueString, valueReal, valueIndicator, valueUnitCode
    FROM A_ClinicalPublicHealthCaseObservationEvent T1
         INNER JOIN A_ClinicalPublicHealthCaseObservationValue value
         ON T1.identifier = value.clinicalObservationEventIdentifier
    WHERE T1.typeCode = @topLevelEventTypeCode
           AND ((@event IS NULL) OR (T1.identifier = @event))
           AND ((@clinicalPublicHealthCaseIdentifier IS NULL)
                OR (T1.clinicalPublicHealthCaseIdentifier = @clinicalPublicHealthCaseIdentifier))
    ) as value
ON event.identifier = value.clinicalObservationEventIdentifier
ORDER BY event.identifier
FOR XML AUTO
```

# Aliased Sub-queries (Event)

---

```
(  
SELECT  
    identifier,                --primary key  
    caseID,                   --foreign key  
    parentObservationEventIdentifier, --parent key  
    sequenceNumber,          --sequence number  
    typeCode                  --type code  
FROM A_ClinicalPublicHealthCaseObservationEvent  
WHERE typeCode = @topLevelEventTypeCode  
    AND ((@event IS NULL) OR (identifier = @event))  
    AND ((@caseID IS NULL) OR (caseID = @caseID))  
) as event
```

# Aliased Sub-queries (Value)

LEFT JOIN

```
(  
    SELECT  
        v.identifier, e.clinicalObservationEventIdentifier,  
        v.sequenceNumber, v.typeCode,  
        v.valueCode, v.valueInteger, v.valueDateTime, v.valueString,  
        v.valueReal, v.valueIndicator, v.valueUnitCode  
    FROM A_ClinicalPublicHealthCaseObservationEvent e  
        INNER JOIN A_ClinicalPublicHealthCaseObservationValue v  
            ON e.identifier = v.clinicalObservationEventIdentifier  
    WHERE e.typeCode = @topLevelEventTypeCode  
        AND ((@event IS NULL) OR (e.identifier = @event))  
        AND ((@caseID IS NULL) OR (e.caseID = @caseID))  
    ) as value
```

# Outer Query

**SELECT**

---

event.identifier, event.parentObservationEventIdentifier,  
event.sequenceNumber, event.typeCode, event.caseID,

value.identifier, value.clinicalObservationEventIdentifier,  
value.sequenceNumber, value.typeCode,  
value.valueCode,  
value.valueInteger,  
value.valueDateTime,  
value.valueString,  
value.valueReal,  
value.valueIndicator,  
value.valueUnitCode

...

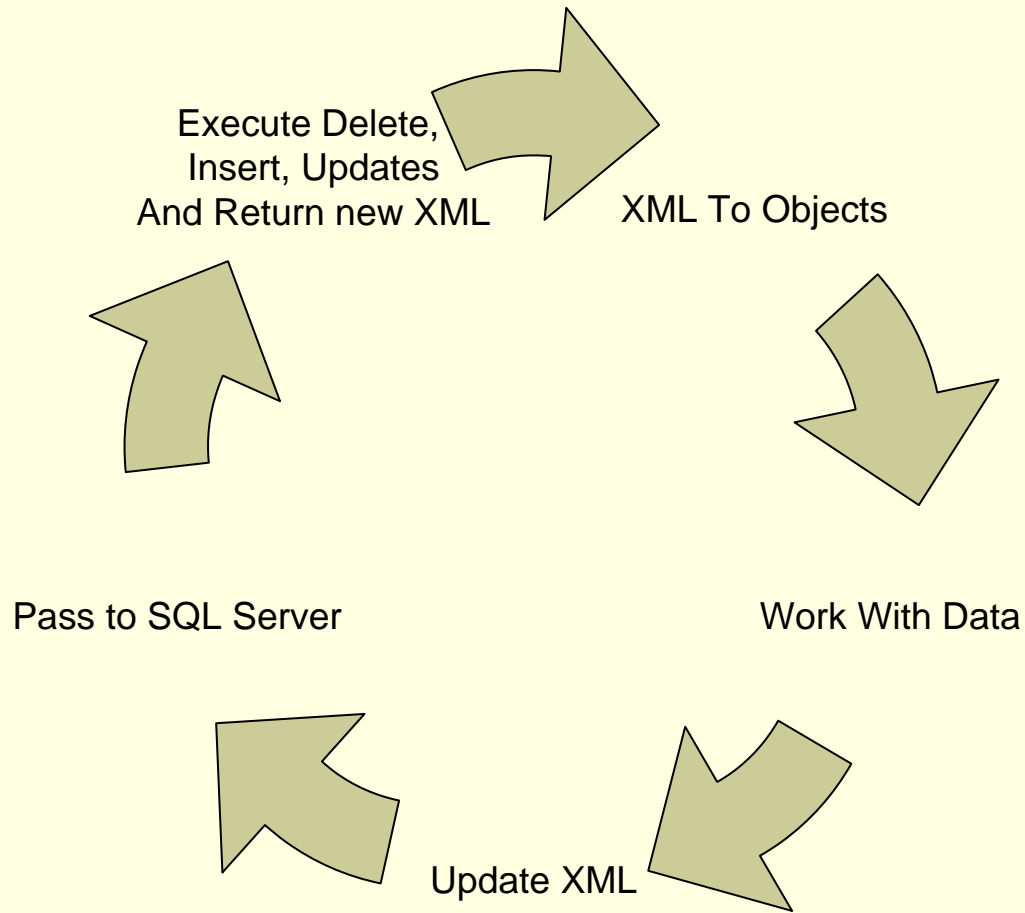
**FOR XML AUTO**

# Observations as XML

```
- <root>
- <event identifier="0D3A8DAC-82CC-44F8-9629-6E8B988EF3A3" typeCode="2655" caseIdentifier="452991">
  <value identifier="64D2A19B-A033-4C48-8981-303E4C78940D" typeCode="2659" valueCode="2900"
    clinicalObservationEventIdentifier="0D3A8DAC-82CC-44F8-9629-6E8B988EF3A3" />
  <value identifier="BE3802C6-9CBA-48B0-8AB4-F63626A0C25D"
    clinicalObservationEventIdentifier="0D3A8DAC-82CC-44F8-9629-6E8B988EF3A3" typeCode="2661"
    valueString="Plastic" />
- <event identifier="462FBE40-CA15-4818-A70B-2AF830E8753F" sequenceNumber="2" typeCode="2656"
  caseIdentifier="452991" parentObservationEventIdentifier="0D3A8DAC-82CC-44F8-9629-6E8B988EF3A3">
  <value identifier="D8DC9C06-887B-4B56-9399-3FE0A14517BF"
    clinicalObservationEventIdentifier="462FBE40-CA15-4818-A70B-2AF830E8753F" typeCode="2657"
    valueString="Ceramic Ladel" />
  <value identifier="8C35FF6B-1E34-4ADD-B8FD-86E673F03A65"
    clinicalObservationEventIdentifier="462FBE40-CA15-4818-A70B-2AF830E8753F" typeCode="2662"
    valueCode="2645" />
  <value identifier="FEEBD0C8-BB90-41FA-ACE9-9FE7FA6F9DCB"
    clinicalObservationEventIdentifier="462FBE40-CA15-4818-A70B-2AF830E8753F" typeCode="2658"
    valueString="Used for serving soups" />
  <value identifier="426ACABA-1DE6-45FC-BC72-FFDD4DE0988B"
    clinicalObservationEventIdentifier="462FBE40-CA15-4818-A70B-2AF830E8753F" typeCode="2660"
    valueCode="2900" />
  </event>
+ <event identifier="3B5B733E-FA43-4083-AE30-A5D576CC8846" sequenceNumber="1" typeCode="2656"
  caseIdentifier="452991" parentObservationEventIdentifier="0D3A8DAC-82CC-44F8-9629-6E8B988EF3A3">
  </event>
</root>
```

# XML In The Application

---



# Saving from XML

```
<event identifier="0D3A8DAC-82CC-44F8-9629-6E8B988EF3A3" typeCode="2655" caseIdentifier="452991">
  <value identifier="64D2A19B-A033-4C48-8981-303E4C78940D" typeCode="2659" valueCode="2900"
    clinicalObservationEventIdentifier="0D3A8DAC-82CC-44F8-9629-6E8B988EF3A3" />
  <value identifier="BE3802C6-9CBA-48B0-8AB4-F63626A0C25D"
    clinicalObservationEventIdentifier="0D3A8DAC-82CC-44F8-9629-6E8B988EF3A3" typeCode="2661"
    valueString="Plastic" />
+ <event identifier="462FBE40-CA15-4818-A70B-2AF830E8753F" sequenceNumber="2" typeCode="2656"
  caseIdentifier="452991" delete="1" parentObservationEventIdentifier="0D3A8DAC-82CC-44F8-9629-
  6E8B988EF3A3">
- <event identifier="3B5B733E-FA43-4083-AE30-A5D576CC8846" sequenceNumber="1" typeCode="2656"
  caseIdentifier="452991" parentObservationEventIdentifier="0D3A8DAC-82CC-44F8-9629-6E8B988EF3A3">
  <value identifier="6C701DCE-DBE4-4F9D-8A37-61FE873706C8"
    clinicalObservationEventIdentifier="3B5B733E-FA43-4083-AE30-A5D576CC8846" typeCode="2658"
    valueString="Drinks hot beverages from the mug" />
  <value identifier="4D99424F-284A-4996-A0D5-AC4488886E42"
    clinicalObservationEventIdentifier="3B5B733E-FA43-4083-AE30-A5D576CC8846" typeCode="2657"
    valueString="Ceramic mug" />
  <value identifier="1B22CED7-7A4D-446D-A8C3-B70615AF42CF"
    clinicalObservationEventIdentifier="3B5B733E-FA43-4083-AE30-A5D576CC8846" typeCode="2662"
    valueCode="2646" />
  <value identifier="F665C6A7-5AAE-4434-BEEA-F5EC236BC54F" delete="1"
    clinicalObservationEventIdentifier="3B5B733E-FA43-4083-AE30-A5D576CC8846" typeCode="2660"
    valueCode="2900" />
  <value identifier="XYZ12345-1ABC-1111-EBAF-A5EC236BC54G" isNew="1"
    clinicalObservationEventIdentifier="3B5B733E-FA43-4083-AE30-A5D576CC8846" typeCode="9999"
    valueCode="2901" />
</event>
</event>
```

# Storing Data Using OpenXML

```
exec sp_xml_preparedocument @doc OUTPUT, @xml

--DELETE Values|
DELETE T1
FROM OpenXML (@doc, '/root//value[@delete="1"]', 1)
    WITH (
        identifier    uniqueidentifier    '@identifier',
        [delete]      bit                  '@delete'
    ) AS oxml
    INNER JOIN A_ClinicalPublicHealthCaseObservationValue T1
        ON oxml.identifier = T1.identifier

DELETE T1
FROM OpenXML (@doc, '/root/event[@delete="1"]', 1)
    WITH (
        identifier    uniqueidentifier    '@identifier',
        [delete]      bit                  '@delete'
    ) AS oxml
    INNER JOIN A_ClinicalPublicHealthCaseObservationEvent T1
        ON oxml.identifier = T1.identifier
```

# Application Code

---

- Created a mini-framework to interact with XML.
- Needs to get a life.

# Conclusions

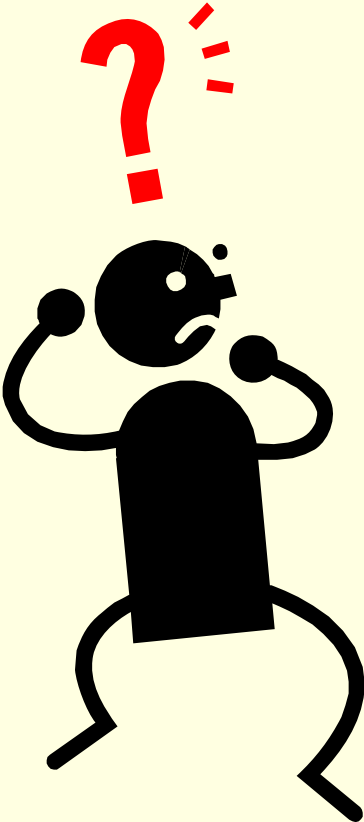
---

- Did we have to do it this way?
  - No
- Were there good reasons for doing it this way?
  - You Bet
- Would we do it again?
  - Probably



# Questions?

---



# Poetry: A Confession

---

Once in a while,  
I'm standing here, doing something.  
And I think,  
"What in the world am I doing here?"  
It's a big surprise.

—*Donald Rumsfeld, May 16, 2001, interview with the New York Times (from The Poetry of D.H. Rumsfeld, by Hart Seely of Slate.com)*

Extra Slides follow . . .

---

# Handling Observations in the App

---

- RIIObservationBase class to handle much of the complexity of working with XML, especially of updating the XML
- Event classes derive from RIIObservationBase and have parent-child relations that match the Event structure.
- Thus, derived classes only implement the following behavior:

# Observation Class Behaviors

---

- Define Enums for the Event and Value TypeCodes they handle
- Implement a PopulateFromXML() method
  - Parent version takes XML doc as parameter
  - Child version takes XML node as parameter
- Implement an UpdateXML() method

# Ceramics Class Enum of Type Codes

---

//ObservationEvent type codes

private enum Events

{

    Ceramics = 2655,     //top level event

    CeramicItem = 2656 //child event

}

//ObservationValue type codes

private enum Values

{

    AreCeramicsUsedInHousehold = 2659,

    UsualFoodDrinkContainer = 2661

}

# Ceramic Item Class Enum of Type Codes

---

```
//ObservationEvent type codes
```

```
private enum Events
```

```
{
```

```
    Ceramics = 2655,    //top level event
```

```
    CeramicItem = 2656 //child event
```

```
}
```

```
//ObservationValue type codes
```

```
private enum Values
```

```
{
```

```
    DescriptionOfItem = 2657,
```

```
    DescriptionOfUse = 2658,
```

```
    WasSampleCollected = 2660,
```

```
    TestKitResult = 2662
```

```
}
```

# Populate From XML Method (Parent)

```
protected override void PopulateFromXml(XmlDocument xml)
{
    XmlNodeList eventNodes = xml.SelectNodes("root/event[@typeCode=\"" +
        Events.Ceramics.ToString() + "\"]");
    if (eventNodes.Count >= 1)
    {
        XmlNode eventNode = eventNodes[0];
        PopulateFromXml(eventNode);
        XmlNodeList ceramicsUsed = xml.SelectNodes("//event[@typeCode=\"" +
            Events.CeramicItem.ToString() + "\"]");
        for (int i = 0; i < ceramicsUsed.Count; i++)
        {
            CeramicUsed ceramicUsed = CeramicUsed.NewCeramicUsed(i);
            ceramicUsed.PopulateFromXml(ceramicsUsed[i]);
            _ceramicsUsed.Add(ceramicUsed);
        }
    }
}
```

# PopulateFromXml (Child Version)

```
protected void PopulateFromXml (XmlNode eventNode)
{
    //Get Event data
    _eventIdIdentifier = new Guid(GetAttributeValue(eventNode, "identifier"));
    _caseIdentifier = GetAttributeValue(eventNode, "caseIdentifier");

    string sIdentifier = _eventIdIdentifier.ToString().ToUpper();
    //Get Value Nodes for parent event
    foreach (XmlNode node in
        eventNode.SelectNodes("//value[@clinicalObservationEventIdentifier='" + sIdentifier + "']"))
    {
        if (node.Attributes["typeCode"] != null)
            switch (Convert.ToInt32(node.Attributes["typeCode"]))
            {
                case (int)Values.AreCeramicsUsedInHousehold:
                    _hasCeramicsInHome = new RInt(GetAttributeValue(node, "valueCode"));
                    break;
                case (int)Values.UsualFoodDrinkContainer:
                    _usualFoodDrinkContainer = GetAttributeValue(node, "valueString");
                    break;
            }
    }
}
```

# UpdateFromXML()

---

- Base class has a SetValue() method to handle receiving values from an object and setting it in the XML doc.
- Top level Observation class (e.g., Ceramics) calls SetValue() for each of its immediate child Observation Values.
- Then triggers each child Event object to do the same.
- Observation Base class prepares the XML sends it to the database for updating.
- Et Voila!

