

How can a Community Member contribute to Pentaho?

Kettle Community Meeting 2019

Leonardo Coelho

Software Engineer – Hitachi Vantara

Jens Bleuel

Sr. Product Manager PDI – Hitachi Vantara

Contribution Overview and Workflow

Pentaho Overview & Technical Tips

Pentaho Marketplace

Summary

Links

Contribution Overview



Code isn't the only way to contribute

There is a very long list of ways to be a contributor:

- QA (Testing)
- Test Cases (unit, integration)
- Use Cases
- Support (forums, how-tos, etc)
- Evangelism
- Defect reporting
- Feature and Enhancement requests
- Social Media (blogging, twitter, etc)
- Documentation
- Translations
- Real world data
- Samples
- Demos
- Graphics and artwork
- Events like this one!
- Meetups
- Access to systems we don't have

And yes, code too.

Code contribution Types

- Bug Fixes
- Unit-tests
- Integration-tests
- Improvements
- Test Automation
- Features
- Cleanup
- Algorithms
- Comments
- Visualizations
- Marketplace Plugins
- Library upgrades
- Big Data Shims

How to get started ?



Quick Jira Primer

1. Search Jira before you create a new case
2. When creating a defect, provide what we need for us to engage:
 - Complete reproduction path
 - Complete logs (SQL/MDX logs if applicable)
 - Screen shots and/or screen cams are very helpful
 - Example demonstrating the issue (schema, ktr/kjb, prpt, etc)
 - Data dump (zipped) if possible
 - Affects version, browser version, OS version, etc.
3. When creating an improvement or feature case, the use case is essential. We have to understand exactly what you're needing to do and why.

Feature and Improvement cases



Feature and improvement Jira cases are triaged and prioritized by our product management team. Part of their job is to review improvement and feature Jira cases, **group them into categories**, and where possible, work them into existing initiatives or Continuous Improvement asks. PM are responsible for determining the overall roadmap for features and improvements and deciding the priority of them.

Defect cases



Defect Jira cases are triaged twice a week by representatives from multiple departments (support, development, quality, and product management).

Once a defect has been through triage, the prioritization team (again, a cross-departmental group) determines the relative priority comparing the defect to other open defects and ranking and grouping it appropriately.

Git Workflow and Coding Standards

Contribution Workflow – Git

1. Fork pentaho/pentaho-kettle
2. Git clone origin/pentaho-kettle
3. Git remote add upstream pentaho/pentaho-kettle
4. Git pull upstream master
5. Git checkout –b PDI-####
6. Do your Magic!
7. Apply Hitachi Vantara Coding Standards





- Hitachi Vantara Coding Standards:
<https://github.com/pentaho/pentaho-coding-standards>

- `System.out.printf("Hello %s", "KCM19");`

Rules: HV Coding Standard

- Checkstyle found 3 item(s) in 1 file(s)
 - RepositoryBrowserController.java : 3 item(s)
 - '(' is not followed by whitespace. (601:23) [ParenPadCheck]
 - ',' is not followed by whitespace. (601:34) [WhitespaceAfterCheck]
 -)' is not preceded with whitespace. (601:41) [ParenPadCheck]

- `System.out.printf("Hello %s", "KCM19");`

Rules: HV Coding Standards

- Checkstyle found no problems in the file(s)

```
/*! *****  
*  
* Pentaho Data Integration  
*  
* Copyright (C) 2002-2019 by Hitachi Vantara : http://www.pentaho.com  
*  
*****
```

Contribution Workflow – Git

8. Git add *

9. Git commit -m “[PDI-#####] <JIRA Case Title>”

10. Git push origin PDI-#####

11. Create a new PR on GitHub:

- pentaho/pentaho-kettle master
- origin/pentaho-kettle PDI-#####



(Quick) Pentaho Overview Technical Tips

PDI / Kettle



PDI / Kettle : Testing a Step

Clone row

Step name Clone row

Nr clones 0

Nr clone in field?

Nr Clone field

Output fields

Add clone flag to output?

Clone flag field

Add clone num to

Clone num field

Help OK Cancel

- Clone row.0 - The field containing the clones number is missing!
- Clone row.0 -
- Clone row.0 - at org.pentaho.di.trans.steps.clonerow.CloneRow.processRow(CloneRow.java:93)

```
public class CloneRowTest {  
    private StepMockHelper<CloneRowMeta, CloneRowData> stepMockHelper;  
  
    @Before  
    public void setup() {  
        stepMockHelper =  
            new StepMockHelper<>( "Test CloneRow", CloneRowMeta.class, CloneRowData.class );  
    }  
  
    @Test( expected = KettleException.class )  
    public void nullNrCloneField() throws Exception {  
        CloneRow step =  
            new CloneRow( stepMockHelper.stepMeta, stepMockHelper.stepDataInterface, 0,  
                stepMockHelper.transMeta, stepMockHelper.trans );  
        step.init( stepMockHelper.initStepMetaInterface, stepMockHelper.initStepDataInterface );  
  
        when( stepMockHelper.processRowsStepMetaInterface.isNrCloneInField() )  
            .thenReturn( true );  
        when( stepMockHelper.processRowsStepMetaInterface.getNrCloneField() )  
            .thenReturn( null );  
  
        step.processRow( stepMockHelper.processRowsStepMetaInterface,  
            stepMockHelper.processRowsStepDataInterface );  
    }  
}
```



```
public class CloneRowTest {  
    private StepMockHelper<CloneRowMeta, CloneRowData> stepMockHelper;  
  
    @Before  
    public void setup() {  
        stepMockHelper =  
            new StepMockHelper<>( "Test CloneRow", CloneRowMeta.class, CloneRowData.class );  
    }  
  
    @Test( expected = KettleException.class )  
    public void nullNrCloneField() throws Exception {  
        CloneRow step =  
            new CloneRow( stepMockHelper.stepMeta, stepMockHelper.stepDataInterface, 0,  
                stepMockHelper.transMeta, stepMockHelper.trans );  
        step.init( stepMockHelper.initStepMetaInterface, stepMockHelper.initStepDataInterface );  
    }  
}
```

```
public class CloneRowTest {  
  private StepMockHelper<CloneRowMeta, CloneRowData> stepMockHelper;  
  
  @Before  
  public void setup() {  
    stepMockHelper =  
      new StepMockHelper<>( "Test CloneRow", CloneRowMeta.class, CloneRowData.class );  
  }  
  
  @Test( expected = KettleException.class )  
  public void nullNrCloneField() throws Exception {  
    CloneRow step =  
      new CloneRow( stepMockHelper.stepMeta, stepMockHelper.stepDataInterface, 0,  
        stepMockHelper.transMeta, stepMockHelper.trans );  
    step.init( stepMockHelper.initStepMetaInterface, stepMockHelper.initStepDataInterface );  
  
    when( stepMockHelper.processRowsStepMetaInterface.isNrCloneInField() )  
      .thenReturn( true );  
    when( stepMockHelper.processRowsStepMetaInterface.getNrCloneField() )  
      .thenReturn( null );  
  
  }  
}
```

```
public class CloneRowTest {  
    private StepMockHelper<CloneRowMeta, CloneRowData> stepMockHelper;  
  
    @Before  
    public void setup() {  
        stepMockHelper =  
            new StepMockHelper<>( "Test CloneRow", CloneRowMeta.class, CloneRowData.class );  
    }  
  
    @Test( expected = KettleException.class )  
    public void nullNrCloneField() throws Exception {  
        CloneRow step =  
            new CloneRow( stepMockHelper.stepMeta, stepMockHelper.stepDataInterface, 0,  
                stepMockHelper.transMeta, stepMockHelper.trans );  
        step.init( stepMockHelper.initStepMetaInterface, stepMockHelper.initStepDataInterface );  
  
        when( stepMockHelper.processRowsStepMetaInterface.isNrCloneInField() )  
            .thenReturn( true );  
        when( stepMockHelper.processRowsStepMetaInterface.getNrCloneField() )  
            .thenReturn( null );  
  
        step.processRow( stepMockHelper.processRowsStepMetaInterface,  
            stepMockHelper.processRowsStepDataInterface );  
    }  
}
```

PDI / Kettle : Testing a Transformation



Generate rows

Step name:

Limit:

Never stop generating rows:

Interval in ms (delay):

Current row time field name:

Previous row time field name:

Fields :

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Value	Set empty string?
1	aField	String							v1	N

```
public class RowGeneratorUnitTest {
    @ClassRule
    public static RestorePDIEngineEnvironment env = new RestorePDIEngineEnvironment();

    @Test
    public void testSafeStop() throws KettleException {
        String filename = getClass().getResource( "safe-stop.ktr" ).getPath();
        TransMeta transMeta = new TransMeta( filename );
        Trans trans = new Trans( transMeta );

        trans.prepareExecution( new String[] { } );
        trans.getSteps().get( 1 ).step.addRowListener( new RowAdapter() {
            @Override
            public void rowWrittenEvent( RowMetaInterface rowMeta, Object[] row ) {
                trans.safeStop();
            }
        } );
        trans.startThreads();
        trans.waitUntilFinished();

        assertEquals( 1, trans.getSteps().get( 0 ).step.getLinesWritten() );
        assertEquals( 1, trans.getSteps().get( 1 ).step.getLinesRead() );
    }
}
```

```
public class RowGeneratorUnitTest {  
    @ClassRule  
    public static RestorePDIEngineEnvironment env = new RestorePDIEngineEnvironment();  
  
    @Test  
    public void testSafeStop() throws KettleException {  
        String filename = getClass().getResource( "safe-stop.ktr" ).getPath();  
        TransMeta transMeta = new TransMeta( filename );  
        Trans trans = new Trans( transMeta );  
  
    }  
}
```

```
public class RowGeneratorUnitTest {  
    @ClassRule  
    public static RestorePDIEngineEnvironment env = new RestorePDIEngineEnvironment();  
  
    @Test  
    public void testSafeStop() throws KettleException {  
        String filename = getClass().getResource( "safe-stop.ktr" ).getPath();  
        TransMeta transMeta = new TransMeta( filename );  
        Trans trans = new Trans( transMeta );  
  
        trans.prepareExecution( new String[] { } );  
        trans.startThreads();  
        trans.waitUntilFinished();  
  
    }  
}
```

```
public class RowGeneratorUnitTest {
    @ClassRule
    public static RestorePDIEngineEnvironment env = new RestorePDIEngineEnvironment();

    @Test
    public void testSafeStop() throws KettleException {
        String filename = getClass().getResource( "safe-stop.ktr" ).getPath();
        TransMeta transMeta = new TransMeta( filename );
        Trans trans = new Trans( transMeta );

        trans.prepareExecution( new String[] { } );
        trans.getSteps().get( 1 ).step.addRowListener( new RowAdapter() {
            @Override
            public void rowWrittenEvent( RowMetaInterface rowMeta, Object[] row ) {
                trans.safeStop();
            }
        } );
        trans.startThreads();
        trans.waitUntilFinished();

    }
}
```



```
public class RowGeneratorUnitTest {
    @ClassRule
    public static RestorePDIEngineEnvironment env = new RestorePDIEngineEnvironment();

    @Test
    public void testSafeStop() throws KettleException {
        String filename = getClass().getResource( "safe-stop.ktr" ).getPath();
        TransMeta transMeta = new TransMeta( filename );
        Trans trans = new Trans( transMeta );

        trans.prepareExecution( new String[] { } );
        trans.getSteps().get( 1 ).step.addRowListener( new RowAdapter() {
            @Override
            public void rowWrittenEvent( RowMetaInterface rowMeta, Object[] row ) {
                trans.safeStop();
            }
        } );
        trans.startThreads();
        trans.waitUntilFinished();

        assertEquals( 1, trans.getSteps().get( 0 ).step.getLinesWritten() );
        assertEquals( 1, trans.getSteps().get( 1 ).step.getLinesRead() );
    }
}
```

- Variable Space:
 - Kettle-variables.xml: **<kettle-variable>**
 - <description>** Description text **</description>**
 - <variable>** EXPECTED_BEHAVIOUR **</variable>**
 - <default-value>** New Behaviour **</default-value>****</kettle-variable>**

- Variable Space:

- Kettle-variables.xml:

```
<kettle-variable>  
  <description> Description text </description>  
  <variable> EXPECTED_BEHAVIOUR </variable>  
  <default-value> New Behaviour </default-value>  
</kettle-variable>
```

- Const.java:

```
public static final String EXPECTED_BEHAVIOUR = "EXPECTED_BEHAVIOUR";
```


- Step Logic:

```
if( this.getVariable(Const.EXPECTED_BEHAVIOUR).equals("New Behaviour") ) {  
  performNewBehaviour();  
} else {  
  performOldBehaviour();  
}
```

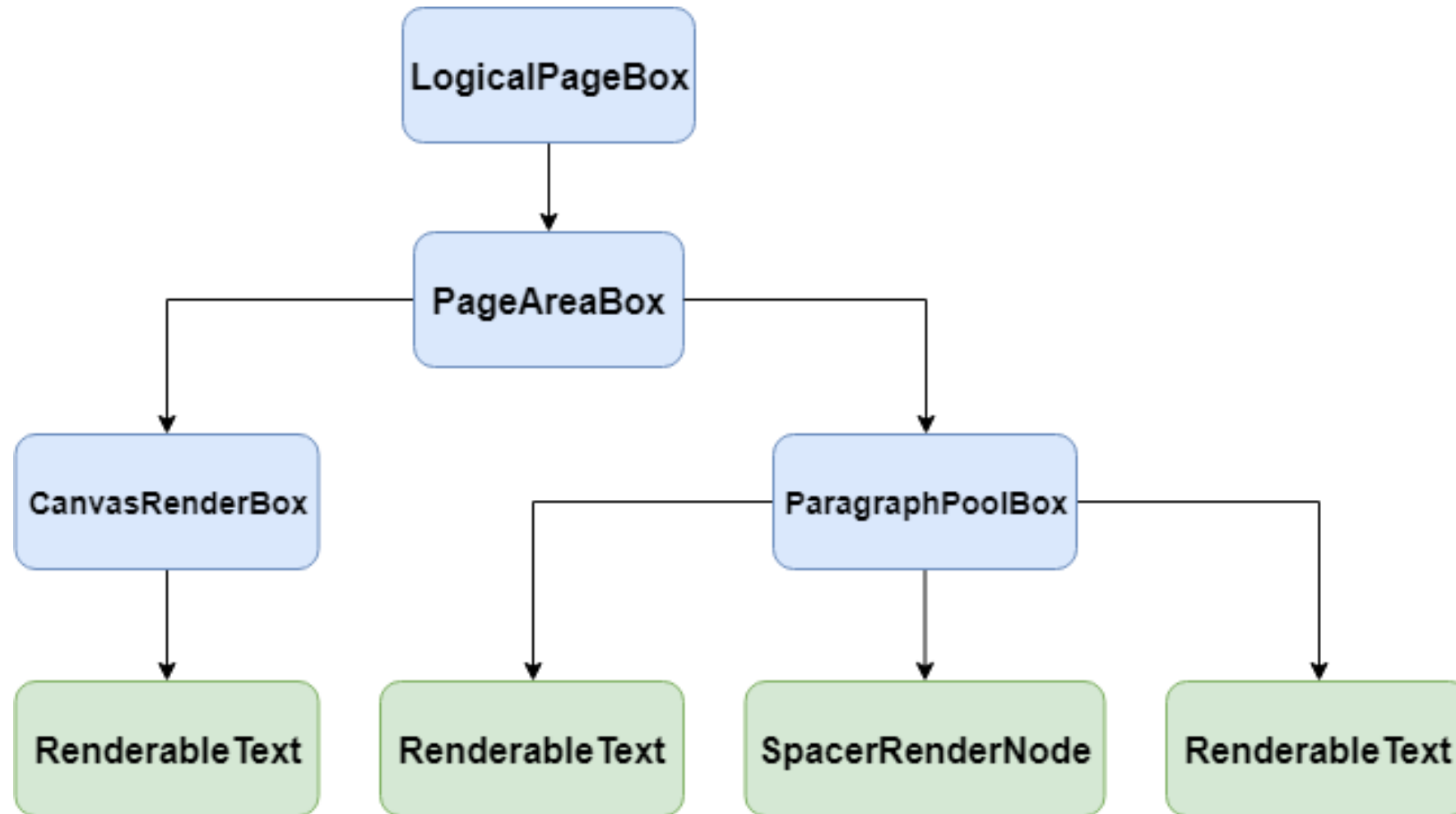
Pentaho Reporting



Pentaho Reporting : Report Page

Steel  Wheels		Steel Wheels, Inc. Income Statement From June 1 through June 30, 2005
Revenue		
Direct Sales	400,000	
Channel Sales	150,000	
Total Revenue		\$ 550,000
Beginning inventory	40,000	
Net purchases	325,000	
Ending inventory	35,000	
Gross Margin		\$ 330,000
Cost of goods sold		
Expenses		
Selling expenses		
Sales salaries	48,000	
Nonrecurring item	12,000	
Other	13,000	
Total Selling expenses		73,000
General and administrative expenses		
Office salaries	27,100	
Depreciation	5,500	
Amortization	3,200	
Bad debt	4,500	
Other	24,200	
Total General and administrative expenses		64,500
Total Expenses		\$ 137,500
Other revenues		
Interest	5,200	
Dividends	7,200	
Gain on sale of equipment	8,600	
Total Other revenues		\$ 21,000
Other expenses		
Interest	9,400	
Writeoff - goodwill	5,000	
Unusual item - loss on sale of long-term investment	5,100	
Total Other expenses		\$ 19,500
Income tax expense	33,600	
Extraordinary item - gain on disposal of business segment	24,000	
Other Comprehensive Income	12,000	
Net Income		\$ 86,400
Report Run: Fri Jul 26 11:22:06 BST 2019		

Pentaho Reporting : Render Nodes Tree



Pentaho Reporting : Testing a Report

```
public class ReportImageTest extends TestCase {  
  protected void setUp() { ClassicEngineBoot.getInstance().start(); }  
  
  public void testGoldRun8() throws Exception {  
    MasterReport report = DebugReportRunner  
      .parseGoldenSampleReport( "Income Statement.prpt" );  
  
    final LogicalPageBox pageBox = DebugReportRunner.layoutPage( report, 0 );  
    // ModelPrinter.INSTANCE.print(pageBox);  
  
    final RenderNode img = MatchFactory.findElementByName( pageBox, "image" );  
    assertEquals( StrictGeomUtility.toInternalValue( 0 ), img.getX() );  
    assertEquals( StrictGeomUtility.toInternalValue( 270 ), img.getWidth() );  
  }  
}
```

Pentaho Reporting : Testing a Report

```
public class ReportImageTest extends TestCase {  
  protected void setUp() { ClassicEngineBoot.getInstance().start(); }  
  
  public void testGoldRun8() throws Exception {  
    MasterReport report = DebugReportRunner  
      .parseGoldenSampleReport( "Income Statement.prpt" );  
  }  
}
```

```
}  
}
```

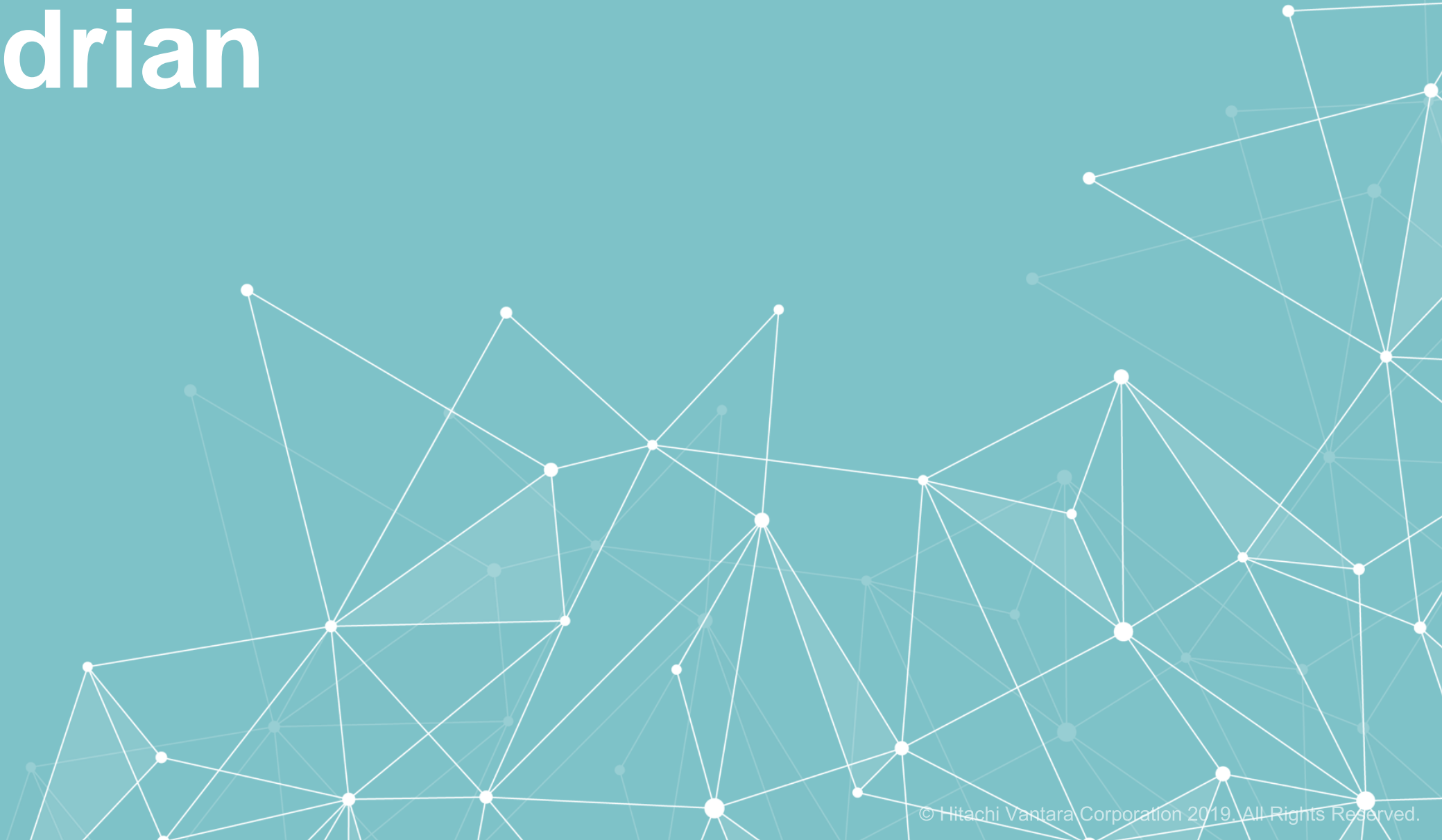

Pentaho Reporting : Testing a Report

```
public class ReportImageTest extends TestCase {  
  protected void setUp() { ClassicEngineBoot.getInstance().start(); }  
  
  public void testGoldRun8() throws Exception {  
    MasterReport report = DebugReportRunner  
      .parseGoldenSampleReport( "Income Statement.prpt" );  
  
    final LogicalPageBox pageBox = DebugReportRunner.layoutPage( report, 0 );  
    // ModelPrinter.INSTANCE.print(pageBox);  
  
  }  
}
```

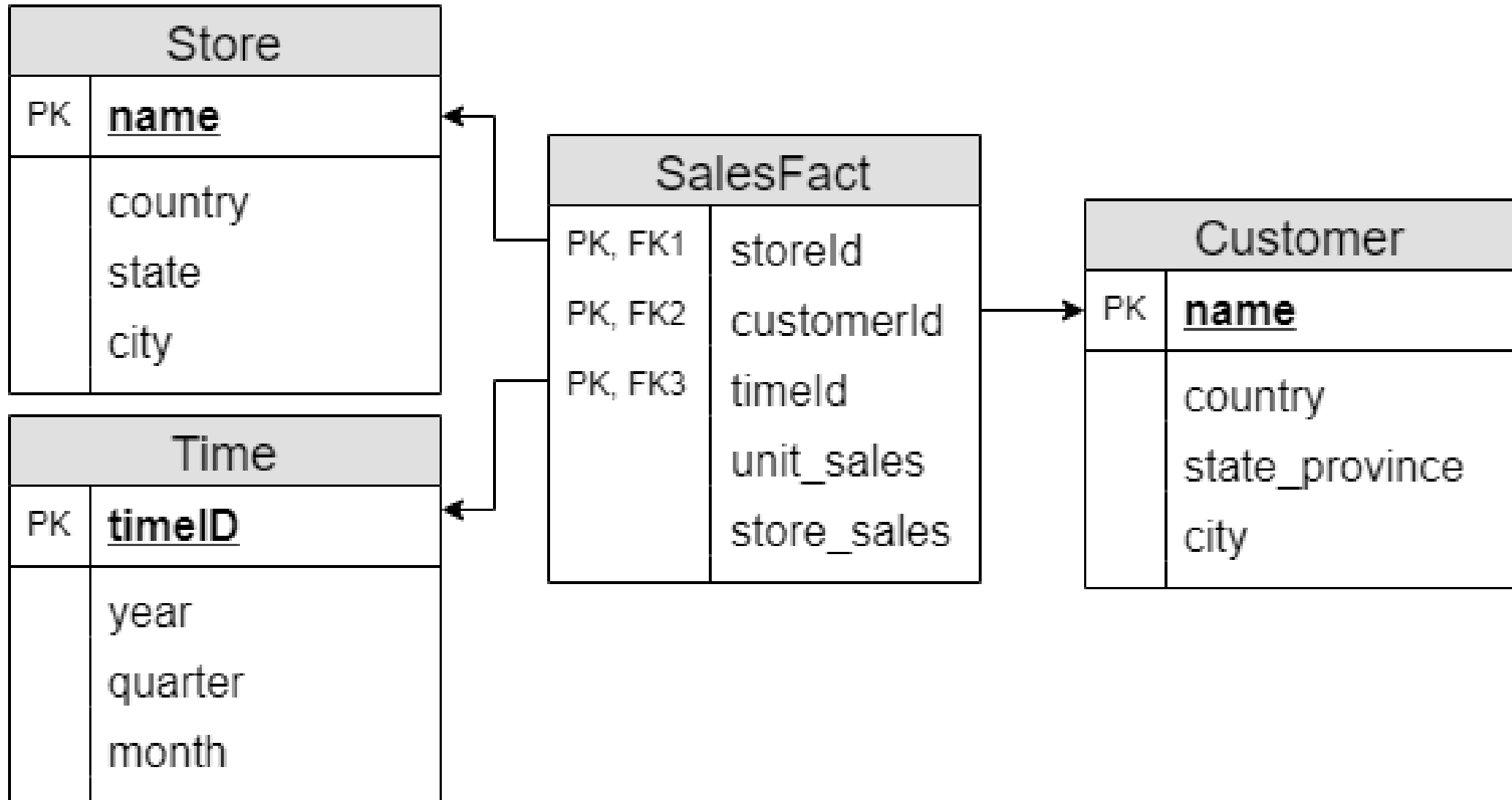
Pentaho Reporting : Testing a Report

```
public class ReportImageTest extends TestCase {  
  protected void setUp() { ClassicEngineBoot.getInstance().start(); }  
  
  public void testGoldRun8() throws Exception {  
    MasterReport report = DebugReportRunner  
      .parseGoldenSampleReport( "Income Statement.prpt" );  
  
    final LogicalPageBox pageBox = DebugReportRunner.layoutPage( report, 0 );  
    // ModelPrinter.INSTANCE.print(pageBox);  
  
    final RenderNode img = MatchFactory.findElementByName( pageBox, "image" );  
    assertEquals( StrictGeomUtility.toInternalValue( 0 ), img.getX() );  
    assertEquals( StrictGeomUtility.toInternalValue( 270 ), img.getWidth() );  
  }  
}
```

Mondrian

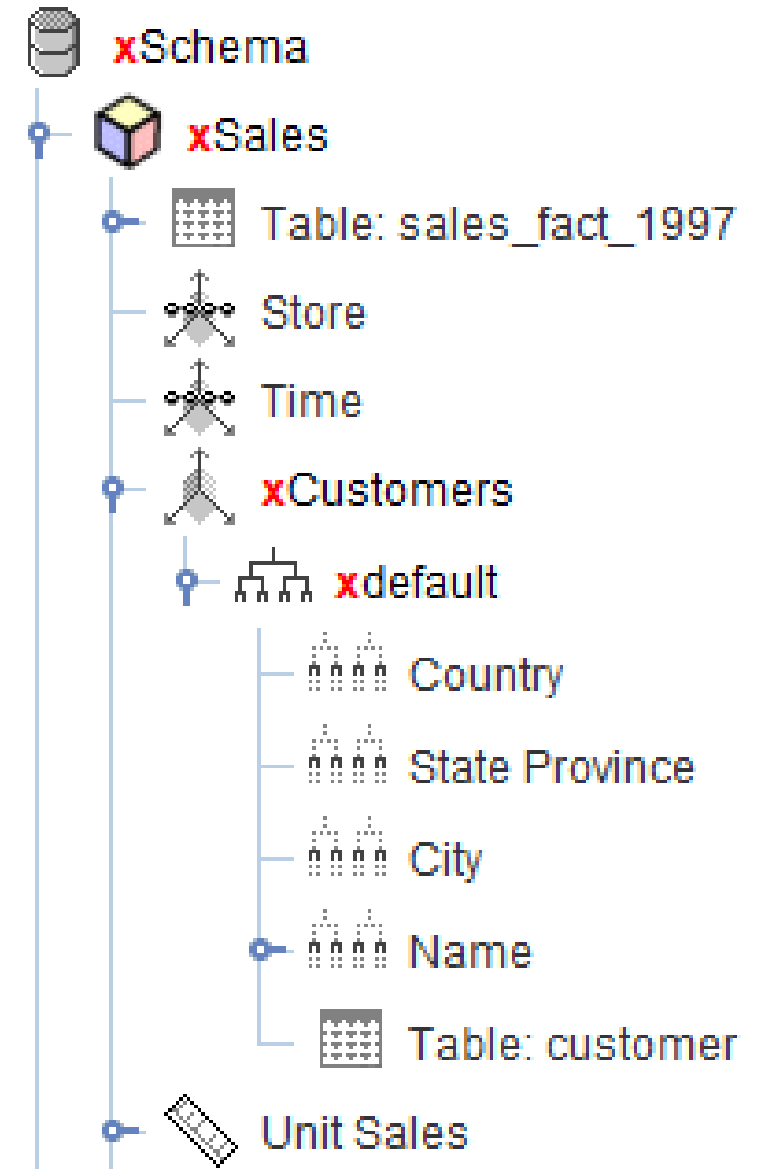


Pentaho Mondrian : Physical Model



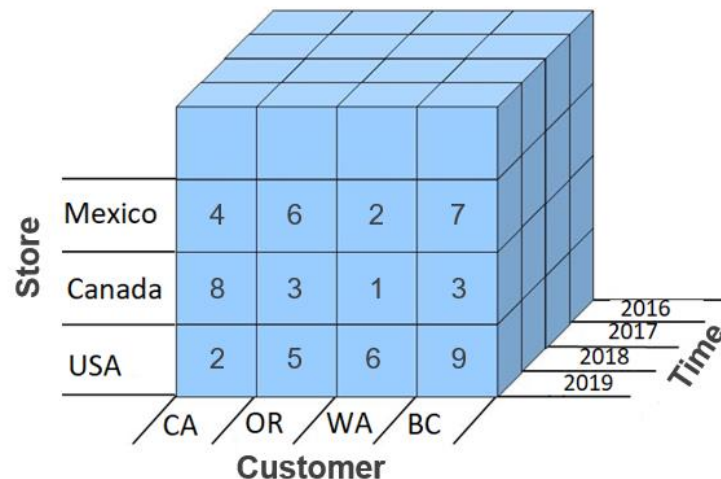
Pentaho Mondrian : Logical Model

- **Cube:** collection of measures and dimensions.
- **Measure:** a quantity value.
- **Dimension:** a set of attributes, by which you can divide measures into sub-categories.
- **Hierarchy:** logical tree structure that defines parent-child relationships.
- **Level:** set of members at the same level.
- **Member:** an attribute of the dimension.



- **MDX** is a query language for OLAP.

```
SELECT
  { [Store].[USA] } ON COLUMNS,
  { [Customers].[USA].children } ON ROWS
FROM [Sales]
WHERE [Time].[1997]
```



	[STORE].[USA]
[CUSTOMERS].[USA].[CA]	24.748
[CUSTOMERS].[USA].[OR]	67.659
[CUSTOMERS].[USA].[WA]	124.366

```
public class SteelWheelsSchemaTest extends SteelWheelsTestCase {  
  public void testMondrian2652() {  
    TestContext testContext = getTestContext();  
    if ( !testContext.databaseIsValid() ) { return; }  
  
    final String schema = "...";  
    final String mdxQuery = "...";  
  
    TestContext testContextReportAuthor =  
      createContext( TestContext.instance(), schema )  
      .withCube( "SteelWheelsSales" ).withRole( "Report Author" );  
  
    testContextReportAuthor.assertQueryThrows( mdxQuery, "MDX object  
'[Dimension2].[All Customers].[Alpha Cognac]' not found in cube 'rolesTest'" );  
  
    TestContext testContextAdministrator =  
      createContext( TestContext.instance(), schema )  
      .withCube( "SteelWheelsSales" ).withRole( "Administrator" );  
  
    testContextAdministrator.assertQueryReturns( mdxQuery, "<<Expected Result>>" );  
  }  
}
```

```
public class SteelWheelsSchemaTest extends SteelWheelsTestCase {  
  public void testMondrian2652() {  
    TestContext testContext = getTestContext();  
    if ( !testContext.databaseIsValid() ) { return; }  
  }  
}
```

```
}  
}
```



```
public class SteelWheelsSchemaTest extends SteelWheelsTestCase {  
  public void testMondrian2652() {  
    TestContext testContext = getTestContext();  
    if ( !testContext.databaseIsValid() ) { return; }  
  
    final String schema = "...";  
    final String mdxQuery = "...";  
  }  
}
```

```
}  
}
```

```
public class SteelWheelsSchemaTest extends SteelWheelsTestCase {  
  public void testMondrian2652() {  
    TestContext testContext = getTestContext();  
    if ( !testContext.databaseIsValid() ) { return; }  
  
    final String schema = "...";  
    final String mdxQuery = "...";  
  
    TestContext testContextReportAuthor =  
      createContext( TestContext.instance(), schema )  
      .withCube( "SteelWheelsSales" ).withRole( "Report Author" );  
  
    testContextReportAuthor.assertQueryThrows( mdxQuery, "MDX object  
'[Dimension2].[All Customers].[Alpha Cognac]' not found in cube 'rolesTest'" );  
  
  }  
}
```

```
public class SteelWheelsSchemaTest extends SteelWheelsTestCase {  
  public void testMondrian2652() {  
    TestContext testContext = getTestContext();  
    if ( !testContext.databaseIsValid() ) { return; }  
  
    final String schema = "...";  
    final String mdxQuery = "...";  
  
    TestContext testContextReportAuthor =  
      createContext( TestContext.instance(), schema )  
      .withCube( "SteelWheelsSales" ).withRole( "Report Author" );  
  
    testContextReportAuthor.assertQueryThrows( mdxQuery, "MDX object  
'[Dimension2].[All Customers].[Alpha Cognac]' not found in cube 'rolesTest'" );  
  
    TestContext testContextAdministrator =  
      createContext( TestContext.instance(), schema )  
      .withCube( "SteelWheelsSales" ).withRole( "Administrator" );  
  
    testContextAdministrator.assertQueryReturns( mdxQuery, "<<Expected Result>>" );  
  }  
}
```

- MondrianProperties.xml:

<PropertyDefinition>

<Name>AlertNativeEvaluationUnsupported**</Name>**

<Path>mondrian.native.unsupported.alert**</Path>**

<Category>SQL generation**</Category>**

<Description>

Alerting action to take in case native evaluation of a function is enabled but not supported for that function's usage in a particular query.

</Description>

<Type>String**</Type>**

<Default>OFF**</Default>**

</PropertyDefinition>

- MondrianProperties.xml:

<**PropertyDefinition**>

<**Name**>AlertNativeEvaluationUnsupported</**Name**>

<**Path**>mondrian.native.unsupported.alert</**Path**>

<**Category**>SQL generation</**Category**>

<**Description**>

Alerting action to take in case native evaluation of a function is enabled but not supported for that function's usage in a particular query.

</**Description**>

<**Type**>String</**Type**>

<**Default**>OFF</**Default**>

</**PropertyDefinition**>

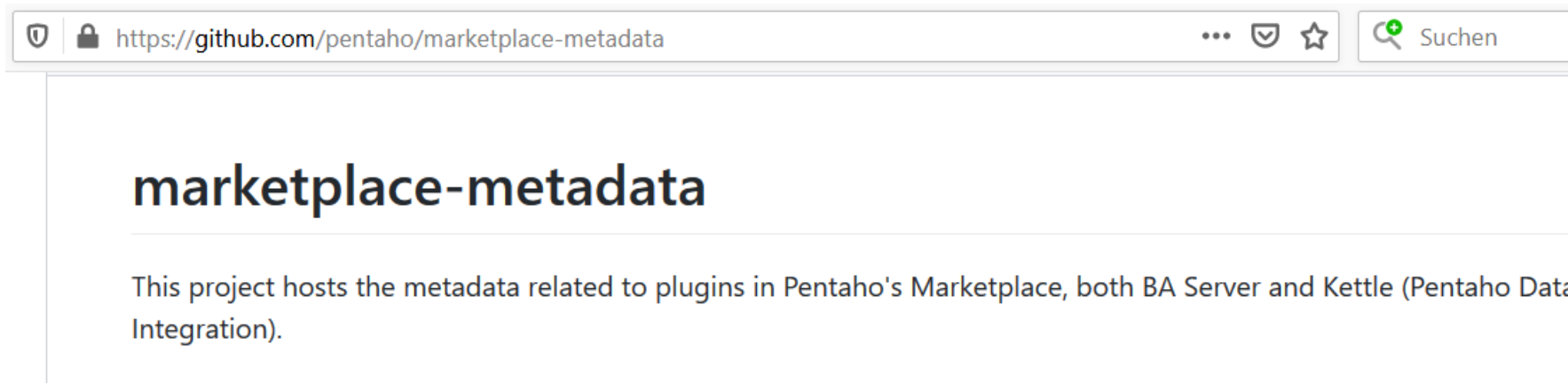
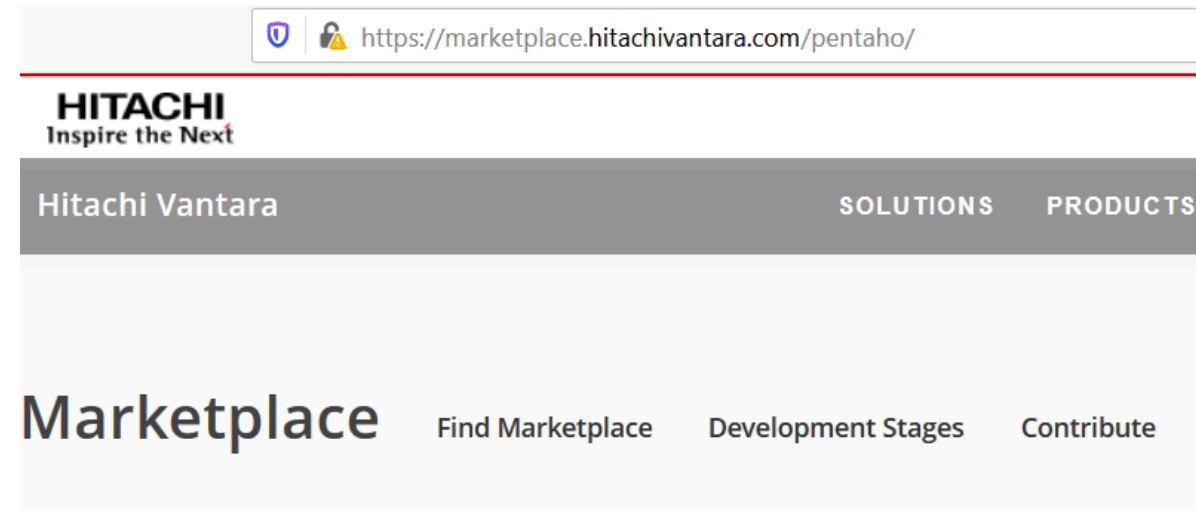
- Update MondrianProperties.java using *mvn package*
- `MondrianProperties.instance().AlertNativeEvaluationUnsupported.get()`

Pentaho Marketplace



Marketplace for New Features

- If you want your feature in the hands of the community quickly, use the marketplace!



Beware of dependencies

Third Party Libraries

- License compatibility (OSS, proprietary)
- Encryption algorithms and export control
- IP auditing

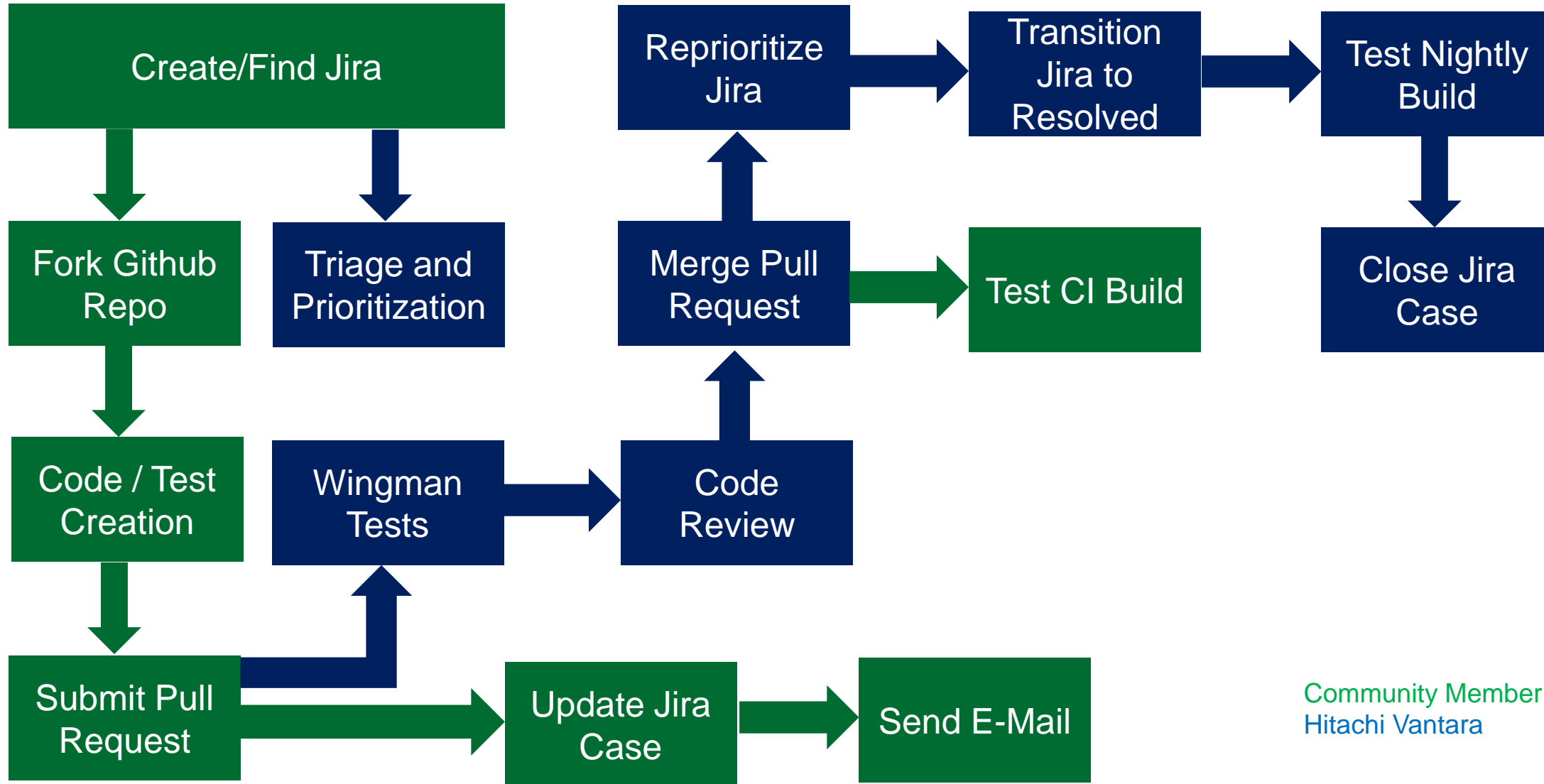
Inter-project Libraries

- License compatibility
- Don't break existing separations between projects and modules

Summary



Code Contribution Flow



Community Member
Hitachi Vantara

Community Member

- Find or create Jira case
- Fork the applicable repository in Github
- Check out Master branch to debug and fix
- Shore up unit test cases for the defect area if necessary
- Fix defect in your fork and add test cases to cover your fix
- Check in change to your fork
- Submit a pull request (Rebase and Squash, no Merge commits)
- Add a comment to the Jira with the PR (and add “patch” to the label field)
- Send e-mail with the Jira case and the PR to community.pentaho@hitachivantara.com
- After we merge it in, we need you to test it on a CI build

Hitachi Vantara / Pentaho

- Triage for severity and prioritize.
- As a pull request comes in, we have a Wingman server that will check the commit for checkstyle, test code coverage comparison from base branch, and will run all the tests to make sure they still run with the same results as before.
- We will code-review the PR, and comment the PR and/or Jira case as necessary
- We will re-prioritize defect Jira cases with patches higher, so they get processed while relevant (within capacity).
- Assuming all is good with the PR, we will merge the PR, and transition the JIRA case as resolved.
- Our QA resources will test the fix in a nightly build and transition the case to closed.

Links



Pentaho

Welcome to the Pentaho section of the Hitachi community. Pentaho's open source heritage has fostered a strong and active community over the past (almost) 13 years and we are excited to have a new home with some impressive interactive features! Please engage, ask questions and explore the power of Pentaho!



HAVE A QUESTION?

Post your question in one of the Pentaho Community Forums below.

- Pentaho
- Pentaho Server
- Data Integration
- Hitachi Content Platform
- Reporting and Analytics
- CTools
- Big Data

Pentaho 8.3

Improve data visibility, accessibility, and agility across edge-to-multicloud environments

LEARN MORE



FOLLOW

NAVIGATION

- Pentaho Community
 - About
 - Pentaho User Groups
 - Downloads
- Projects
 - Mondrian
 - Data Integration - Kettle
 - Pentaho Reporting
 - Data Mining - Weka

CTools

- Community Dashboard Editor
- Community Dashboard



Useful Documentation and Repositories

Link	Description
https://github.com/pentaho/pentaho-coding-standards	Pentaho Coding Standards
https://help.pentaho.com/Documentation/8.3/Developer_center/Embed_and_extend_PDI_functionality	Embed and extend PDI functionality
https://github.com/pentaho/pentaho-engineering-samples	Pentaho Engineering Samples
https://github.com/pentaho/pdi-sdk-plugins	Samples as part of the PDI SDK
https://github.com/pentaho/maven-project-archetypes	For example, if you wanted to create a CE PDI step plugin, you would follow the instructions located in the README here: https://github.com/pentaho/maven-project-archetypes/tree/master/pentaho-pdi-step-plugin-archetype

Main Links

Link	Description
https://community.hitachivantara.com/s/pentaho	Main community web site for Pentaho, including Forums and Articles
https://help.pentaho.com	Pentaho Documentation (some can still be found in https://wiki.pentaho.com)
https://jira.pentaho.com	Where you search for and file defects, improvements, and feature requests
https://github.com/pentaho	Where the source code lives
https://sourceforge.net/projects/pentaho/	Site that has compiled stable binaries
https://nexus.pentaho.org	Artifact repository where things get resolved from
https://help.pentaho.com/Documentation/8.3/Developer_center	APIs, Customize Pentaho Products, Extending and Embedding
https://github.com/pentaho/marketplace-metadata	Pentaho Marketplace Repository (see also https://marketplace.hitachivantara.com/pentaho/)
http://kettle.bleuel.com/2019/11/23/pentaho-community-contributions-kcm19/	All the links in a blog post

Thank You



HITACHI
Inspire the Next 