Evolveum

Native reports

Lukáš Škublík / February 2022

Agenda

- Before 4.4
- Native reports
- Basic parts of Native report
- Report tasks
- Converting of Jasper to Native report

Before 4.4

- Using Jasper framework

- Needed Jaspersoft Studio

- Complicated customization

- Report object contains an additional base64 encoded jasper xml

- Jasper reports were removed in MidPoint 4.4

Native reports

- Using native object of MidPoint

  - Dashboards

  - Object collection

- Simple creating of new reports

- Support

  - multithreading

  - bucketing (only for collection reports of objects and audit records)

- Every report output is store in ‚MidPoint_home'/export

Basic parts of Native report

Evolveum

- Behavior
  - Import
  - Export
- Engine
  - Dashboard
  - Object collection
- File format
  - CSV
  - HTML

```xml
<report>

        <name>Report example</name>

        <behavior>

                <direction>import/export</direction>

        </behavior>

<dashboard/>/<objectCollection/>

<fileFormat>

                <type>csv/html</type>

</fileFormat>

</report>
```

Collection report

- Supported kind of object

  - Audit records

  - Containers

  - Objects in MidPoint (UserType, RoleType, OrgType, …)

- Part of configuration

  - Query

  - Columns
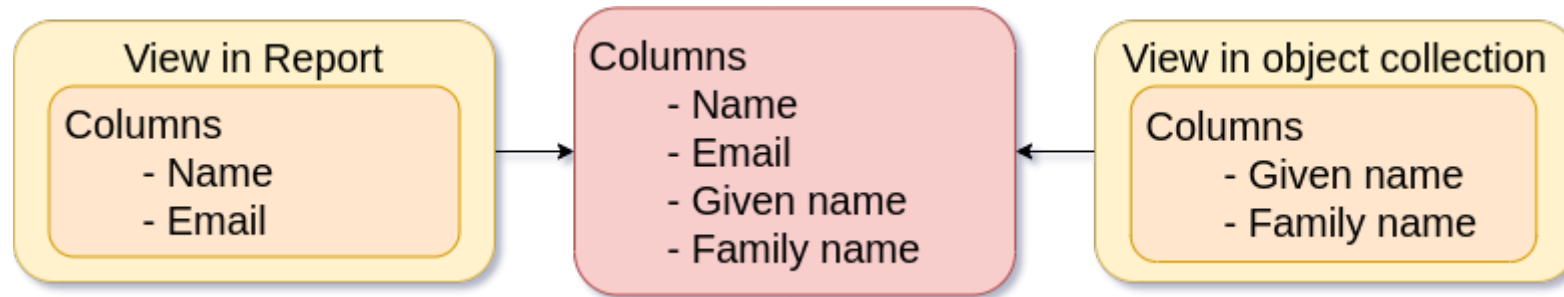
  - Parameters

  - Subreports

  - Condition

Collection report - query

- Attribute *collection*

- Three variants

  - Reference object collection

  - Filter query

  - Filter query with base collection
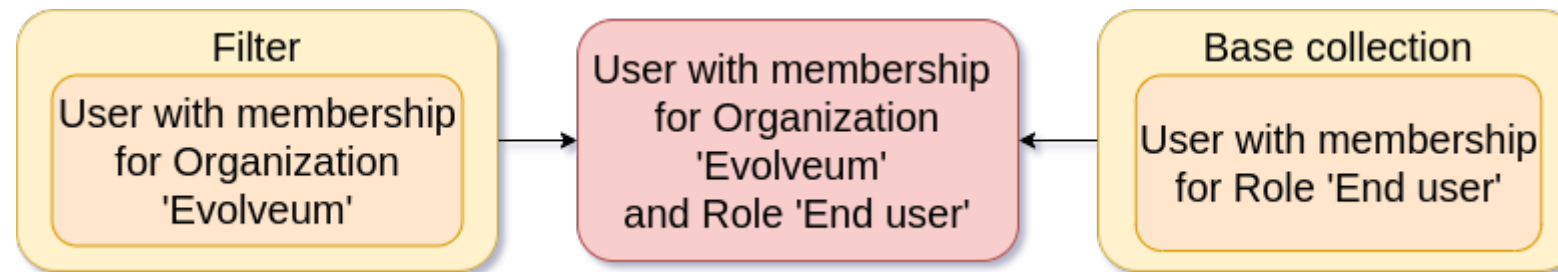
Collection report - columns

- Attribute *view*

- Columns are merged from views

- Can use only view in report – *useOnlyReportView*

- Default view for audit records and MidPoint objects

Collection report - columns

Evolveum

```xml
<view>
    <column>
        <name>nameColumn</name>
        <path>name</path>
    </column>
    <column>
        <name>fullNameColumn</name>
        <path>fullName</path>
        <previousColumn>nameColumn</previousColumn>
    </column>
    <type>UserType</type>
</view>
```

Users in MidPoint (Users in MidPoint)

Back | Save | Save and run | Change archetype | Delete object | Edit raw

Report operations
Show report preview | Show report preview in popup | ▶

Basic
Engine
Collection
File format
View
Parameter
Subreport

Type ⓘ User

Columns

	Name * ⓘ	Path ⓘ		Label ⓘ			
☐	nameColumn ⊖	name	⊖		⊟	⊖	▬ 🖉
		Reset path					
☐	fullNameColumn ⊖	fullName	⊖		⊟	⊖	▬ 🖉
		Reset path					
☐	activationColumr ⊖	activation/administrativeStatus	⊖		⊟	⊖	▬ 🖉
		Reset path					
☐	roleColumn ⊖	assignment	⊖	Role	⊟	⊖	▬ 🖉
		Reset path					
☐	orgColumn ⊖	assignment	⊖	Organization	⊟	⊖	▬ 🖉
		Reset path					
☐	accountColumn	linkRef	⊖	Account		⊖	▬ 🖉

Collection report - parameters

Evolveum

- Attribute *parameter*

```xml
<parameter>
    <name>activation</name>
    <type>ActivationStatusType</type>
    <display>
        <label>
            <t:orig>activation</t:orig>
            <t:norm>activation</t:norm>
            <t:translation>
                <t:key>ActivationType.administrativeStatus</t:key>
            </t:translation>
        </label>
    </display>
</parameter>
```

MidPoint (Users in MidPoint)

Report operations

| 🖫 Save and run | A Change archetype | 🗑 Delete object | ☑ Edit raw | 🖼 Show report preview | 🖼 Sho... |

	Name ℹ	Type ℹ	Label ℹ		
☐	activation	ActivationStatusType ⊖	Administrative status 📋 ⊖		— ☑
☐	organizationRef	ObjectReferenceType ⊖	Organization 📋 ⊖		— ☑
☐	roleRef	ObjectReferenceType ⊖	Role 📋 ⊖		— ☑
☐	resourceRef	ObjectReferenceType ⊖	Resource 📋 ⊖		— ☑

Collection report - parameters

- Attribute *parameter*

Collection report - subreports

- Attribute *subreport*

- Subreport variable is type collection

```xml
<column>
    <name>emailInMpColumn</name>
    <display>
        <label>Email in mP</label>
    </display>
    <previousColumn>lastNameColumn</previousColumn>
    <export>
        <expression>
            <script>
                <code>
                    if (!user.isEmpty()) {
                        return user.get(0).asObjectable().getEmailAddress();
                    }
                </code>
            </script>
        </expression>
    </export>
</column>
```

```xml
<subreport>
    <name>user</name>
    <type>UserType</type>
    <expression>
        <script>
            <code>
                return midpoint.searchShadowOwner(object.getOid());
            </code>
        </script>
    </expression>
</subreport>
```
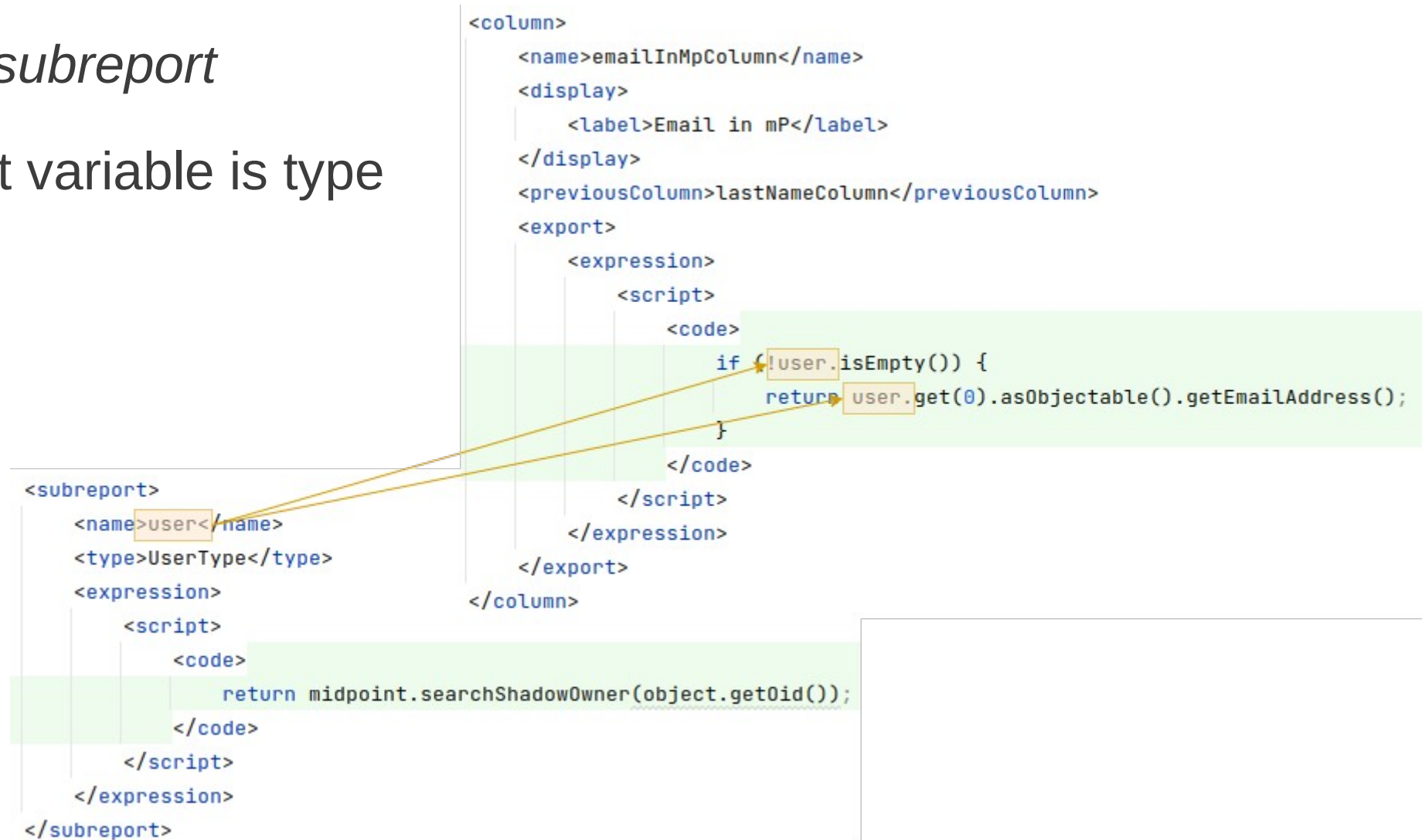
Collection report - condition

- Attribute *condition*

- Use only when query is insufficient

- High performance requirements

Collection report – example

Evolveum

Users in MidPoint

Report generated on: Saturday, 5. Feb 2022 19:13:36

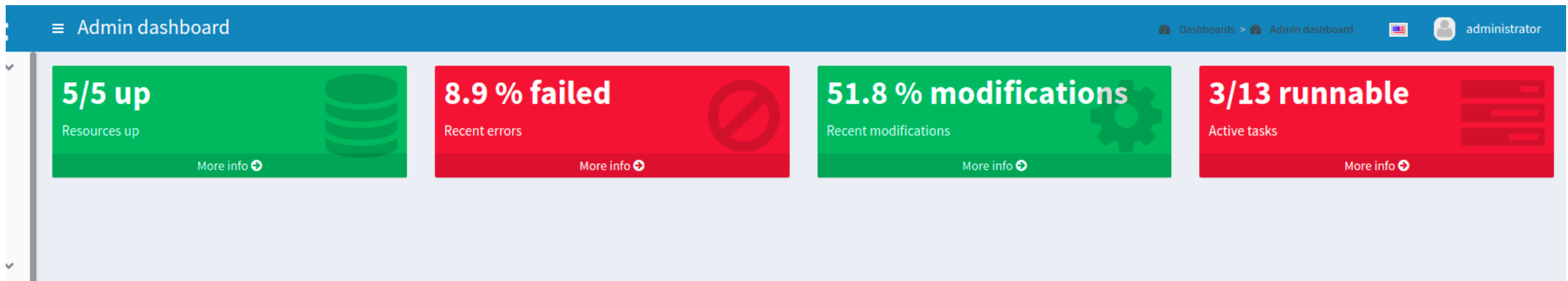| Name	Full name	Administrative status	Role	Organization	Account
admin	Admin Admin	Enabled	Superuser	ADMIN	
administrator	midPoint Administrator	Enabled	Superuser		
janeway	Kathryn Janeway	Enabled	Secret Projects I		kjaneway(Resource: CSV-1 (Document Access))
kirk	Jim Tiberius Kirk	Enabled	Internal Employee CSV-3 Admin CSV-1 Tester	0919	cn=Jim Tiberius Kirk,ou=ExAmPLE,dc=example,dc=com(Resource: CSV-3 (LDAP)) jkirk(Resource: CSV-1 (Document Access)) jkirk(Resource: CSV-2 (Canteen Ordering System)) _kirk(Resource: CSV-1 (Document Access)) cn=Jim Tiberius Kirk,ou=_Administrators_,ou=ExAmPLE,dc=example,dc=com(Resource: CSV-3 (LDAP))
picard	Jean-Luc Picard	Enabled			jpicard(Resource: CSV-1 (Document Access))
X000002	Richard Dwayne	Disabled		FORMER	000002(Resource: ExAmPLE, Inc. HR Source)
X000005	Eve Morthanic	Enabled	Internal Employee Basic user Address Book Attributes Manager Terraforming Terraforming2	ACTIVE 0100 0100	000005(Resource: ExAmPLE, Inc. HR Source) emorthan(Resource: CSV-2 (Canteen Ordering System)) emorthanic(Resource: CSV-1 (Document Access)) cn=Eve Morthanic,ou=0100,ou=ExAmPLE,dc=example,dc=com(Resource: CSV-3 (LDAP)) uid=X000005,ou=people,dc=evolveum,dc=com(Resource: OpenLDAP)
X000021	Rudie Mechal	Enabled	Internal Employee Basic user User Manager	ACTIVE 0100	000021(Resource: ExAmPLE, Inc. HR Source) rmechal(Resource: CSV-2 (Canteen Ordering System)) rmechal(Resource: CSV-1 (Document Access)) cn=Rudie Mechal,ou=0100,ou=ExAmPLE,dc=example,dc=com(Resource: CSV-3 (LDAP))
X000035	James Bradley	Enabled	Internal Employee Basic user	ACTIVE 0200 0200	000035(Resource: ExAmPLE, Inc. HR Source) jbradley(Resource: CSV-2 (Canteen Ordering System)) jbradley(Resource: CSV-1 (Document Access)) cn=James Bradley,ou=0200,ou=ExAmPLE,dc=example,dc=com(Resource: CSV-3 (LDAP))
X000045	Andreas Decker	Enabled	Internal Employee	ACTIVE 0110	000045(Resource: ExAmPLE, Inc. HR Source) adecker(Resource: CSV-2 (Canteen Ordering System))

Collection report – example

- Audit report with changed attributes

- Comparing attributes of shadow and attributes of user in midPoint

- Long unchanged accounts

- Users without account

- All examples are in the docs!

  - https://docs.evolveum.com/midpoint/reference/misc/reports/examples/

Dashboard report

- Create report from dashboard

  - Documentation for dashboard configuration
    https://docs.evolveum.com/midpoint/reference/admin-gui/dashboards/configuration/

- Skip tables for individual widgets by attribute *showOnlyWidgetsTable*

Dashboard report - view

Evolveum

- Attribute *storeExportedWidgetData*

- Possible values:
  - *onlyWidget*
  - *onlyFile*
  - *WidgetAndFile*

- Documentation for asynchronous dashboard configuration
  https://docs.evolveum.com/midpoint/reference/admin-gui/dashboards/configuration/#asynchronous-widget
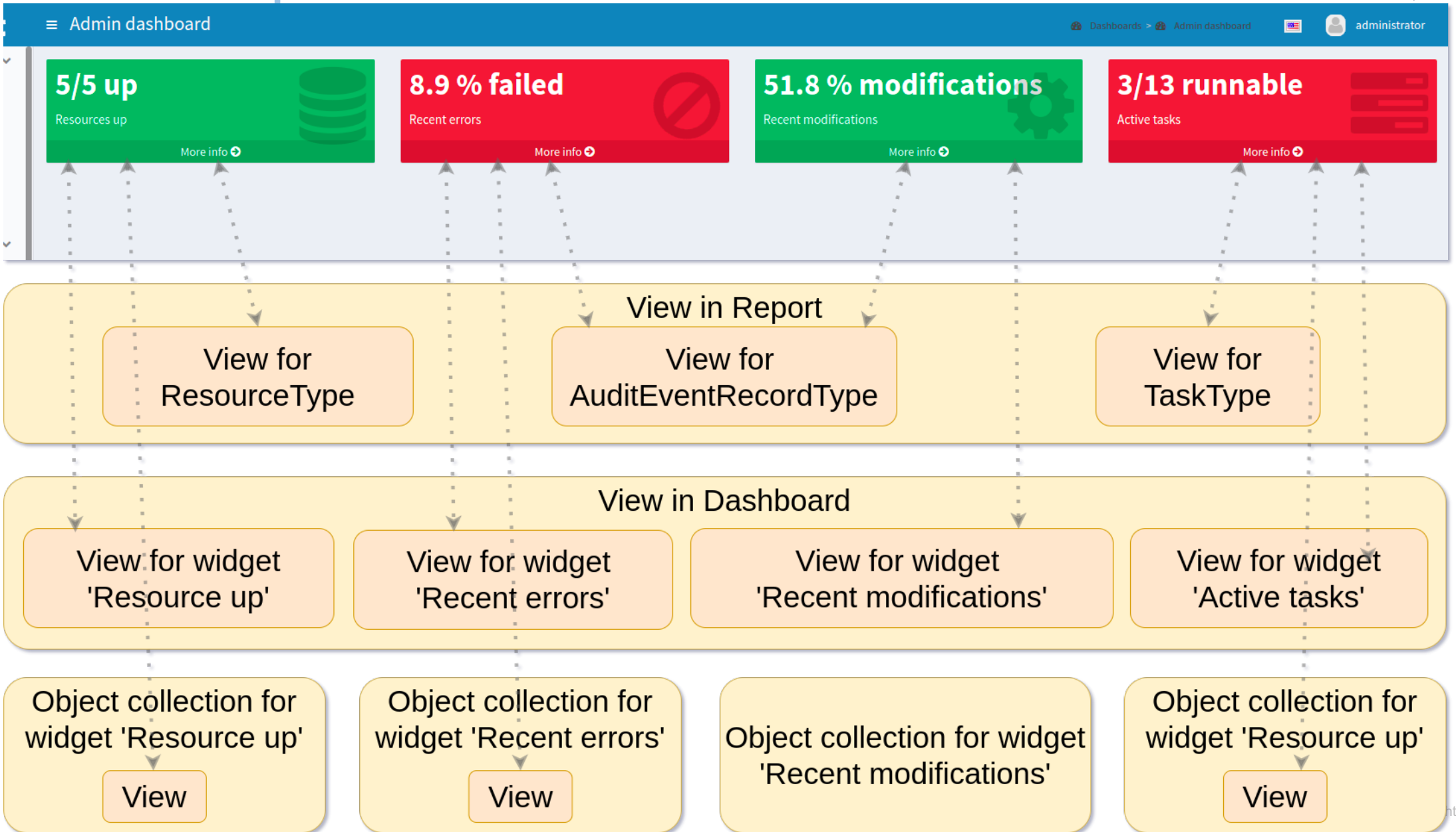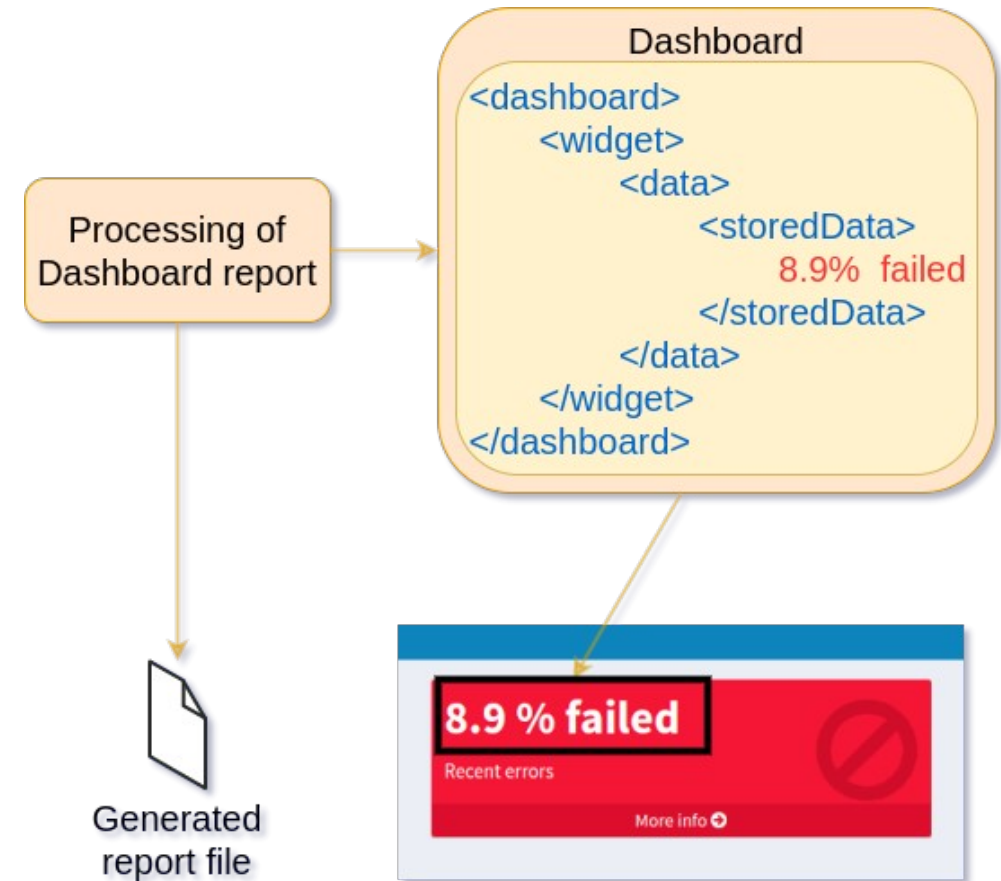
Dashboard report – output

Evolveum

Widget

Report generated on: Thursday, 3. Feb 2022 21:23:37

Label	Number	Status
Resources up	5/5 up	
Recent errors	8.3 % failed	
Recent modifications	50 % modifications	
Active tasks	3/14 runnable	

Resources up

Report generated on: Thursday, 3. Feb 2022 21:23:37

Name	Connector type	Connector version
CSV-1 (Document Access)	com.evolveum.polygon.connector.csv.CsvConnector	2.4
CSV-2 (Canteen Ordering System)	com.evolveum.polygon.connector.csv.CsvConnector	2.4
CSV-3 (LDAP)	com.evolveum.polygon.connector.csv.CsvConnector	2.4
ExAmPLE, Inc. HR Source	com.evolveum.polygon.connector.csv.CsvConnector	2.4
OpenLDAP	com.evolveum.polygon.connector.ldap.LdapConnector	3.3

Recent errors

Report generated on: Thursday, 3. Feb 2022 21:23:37

Time	Initiator	Event stage	Event type	Target	Outcome	Message	Delta

- From 4.4 MidPoint contains three activity work definition for task

  - ClassicReportExportWorkDefinitionType *reportExport*

    - Support multithreading (containers, objects and audit records)

  - ClassicReportImportWorkDefinitionType *reportImport*

    - Support multithreading (only objects)

  - DistributedReportExportWorkDefinitionType *distributedReportExport*

    - Support multithreading and bucketing (objects and audit records)

Convert *reportExport* to *distributedReportExport*

```xml
<task>
    ...
    <name>Report task</name>
    <assignment>
        <targetRef oid="00000000-0000-0000-0000-000000000511"
                relation="org:default"
                type="c:ArchetypeType"/>
    </assignment>
    <archetypeRef oid="00000000-0000-0000-0000-000000000511"
                relation="org:default"
                type="c:ArchetypeType"/>
    <roleMembershipRef oid="00000000-0000-0000-0000-000000000511"
                relation="org:default"
                type="c:ArchetypeType"/>
    <activity>
        <work>
            <reportExport>
                <reportRef oid="00000000-0000-0000-0000-000000000110"
                        relation="org:default"
                        type="c:ReportType">
                    <!-- Users in MidPoint -->
                </reportRef>
            </reportExport>
        </work>
    </activity>
    ...
</task>
```

```xml
<task>
    ...
    <name>Report task</name>
    <assignment>
        <targetRef oid="00000000-0000-0000-0000-000000000512"
                relation="org:default"
                type="c:ArchetypeType"/>
    </assignment>
    <activity>
        <work>
            <distributedReportExport>
                <reportRef oid="00000000-0000-0000-0000-000000000110"
                        relation="org:default"
                        type="c:ReportType">
                    <!-- Users in MidPoint -->
                </reportRef>
            </distributedReportExport>
        </work>
    </activity>
    ...
</task>
```

Demo

Evolveum

Converting of Jasper to Native report - query

```xml
<report oid="00000000-0000-0000-0000-000000000110"
        xmlns="http://midpoint.evolveum.com/xml/ns/public/common/common-3">
    <name>Users in MidPoint</name>
    <description>Users listed in MidPoint.</description>
    <reportEngine>jasper</reportEngine>
    <jasper>
        <parent>true</parent>
        <template>PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4KPGphc3BlclJlc ..... </template>
        <templateStyle>PD94bWwgdmVyc2lvbj0iMS4wIiJ8+DQo8IURPQ1RZUEUgamFzcGVyVGVtcGxh ..... </templateStyle>
        <export>pdf</export>
        <virtualizer>JRSwapFileVirtualizer</virtualizer>
        <virtualizerKickOn>300</virtualizerKickOn>
        <maxPages>10000</maxPages>
        <timeout>300000</timeout>
    </jasper>
</report>
```

Converting of Jasper to Native report - query

```xml
<report oid="00000000-0000-0000-0000-000000000110"
        xmlns="http://midpoint.evolveum.com/xml/ns/public/common/common-3">
    <name>Users in MidPoint</name>
    <description>Users listed in MidPoint.</description>
    <objectCollection>

    </objectCollection>
</report>
```

Converting of Jasper to Native report - query

```xml
<queryString language="mql">
    <![CDATA[<filter>
        <and>
            <equal>
                <path>activation/administrativeStatus</path>
                <expression>
                    <queryInterpretationOfNoValue>filterAll</queryInterpretationOfNoValue>
                    <path>$activation</path>
                </expression>
            </equal>
            <ref>
                <path>assignment/targetRef</path>
                <expression>
                    <queryInterpretationOfNoValue>filterAll</queryInterpretationOfNoValue>
                    <script>
                        <code>
                            import com.evolveum.midpoint.xml.ns._public.common.common_3.*;

                            if (!role) {
                                return null;
                            }

                            ObjectReferenceType ort = new ObjectReferenceType();
                            ort.setOid(role);
                            ort.setType(RoleType.COMPLEX_TYPE);
                            return ort;
                        </code>
                    </script>
                </expression>
            </ref>
        </and>
    </filter>]]>
</queryString>
```

```xml
<filter xmlns="http://midpoint.evolveum.com/xml/ns/public/common/common-3"
        xmlns:q="http://prism.evolveum.com/xml/ns/public/query-3">
<q:and>
    <q:equal>
        <q:path>activation/administrativeStatus</q:path>
        <expression>
            <queryInterpretationOfNoValue>filterAll</queryInterpretationOfNoValue>
            <path>$activation</path>
        </expression>
    </q:equal>

    <q:ref>
        <q:path>assignment/targetRef</q:path>
        <expression>
            <queryInterpretationOfNoValue>filterAll</queryInterpretationOfNoValue>
            <path>$role</path>
        </expression>
    </q:ref>
</q:and>
</filter>
```

- Find elements *jasperReport/parameter*

```
<parameter name="activation" class="com.evolveum.midpoint.xml.ns._public.common.common_3.ActivationStatusType"/>
<parameter name="role" class="java.lang.String"/>


<parameter>
    <name>role</name>
    <type>ObjectReferenceType</type>
    <targetType>RoleType</targetType>
    <display>
        <label>
            <t:orig>role</t:orig>
            <t:norm>role</t:norm>
            <t:translation>
                <t:key>ObjectTypeGuiDescriptor.role</t:key>
            </t:translation>
        </label>
    </display>
</parameter>
```

```
<parameter>
    <name>activation</name>
    <type>ActivationStatusType</type>
    <display>
        <label>
            <t:orig>activation</t:orig>
            <t:norm>activation</t:norm>
            <t:translation>
                <t:key>ActivationType.administrativeStatus</t:key>
            </t:translation>
        </label>
    </display>
</parameter>
```

Evolveum

- Find element *jasperReport/detail/band/frame* and in it the elements *textField* and *componentElement*

  - *textField :*

```
<textField isStretchWithOverflow="true" isBlankWhenNull="true">        <column>
    <reportElement x="0" y="0" width="280" height="20" uuid="3c668eee-cd3e-4697-afe3-edb7894525cc"/>        <name>nameColumn</name>
    <textFieldExpression><![CDATA[$F{name}]]></textFieldExpression>        <path>name</path>
</textField>        </column>
```

Evolveum

- Find element *jasperReport/detail/band/frame* and in it the elements *textField* and *componentElement*

  - *componentElement :*

```
<componentElement>
    ...
    <jr:list>
        <datasetRun subDataset="rolesDataset" uuid="300586d6-0f32-4af9-b27f-ab99167501d5">
            <parametersMapExpression><![CDATA[new HashMap($P{REPORT_PARAMETERS_MAP})]]></paramet
            <datasetParameter name="assignment">
                <datasetParameterExpression><![CDATA[$F{assignment}]]></datasetParameterExpressi
            </datasetParameter>
            <connectionExpression><![CDATA[$P{REPORT_CONNECTION}]]></connectionExpression>
        </datasetRun>
        ...
    </jr:list>
</componentElement>

<subDataset name="rolesDataset" uuid="659d3fba-d03d-43bc-8dcd-502d03443ebe">
    <parameter name="assignment" class="java.util.List"/>
    <queryString language="mql">
        <![CDATA[<code>if (assignment != null){report.resolveRoles(assignment)}</code>]]>
    </queryString>
    <field name="name" class="java.lang.String"/>
</subDataset>
```
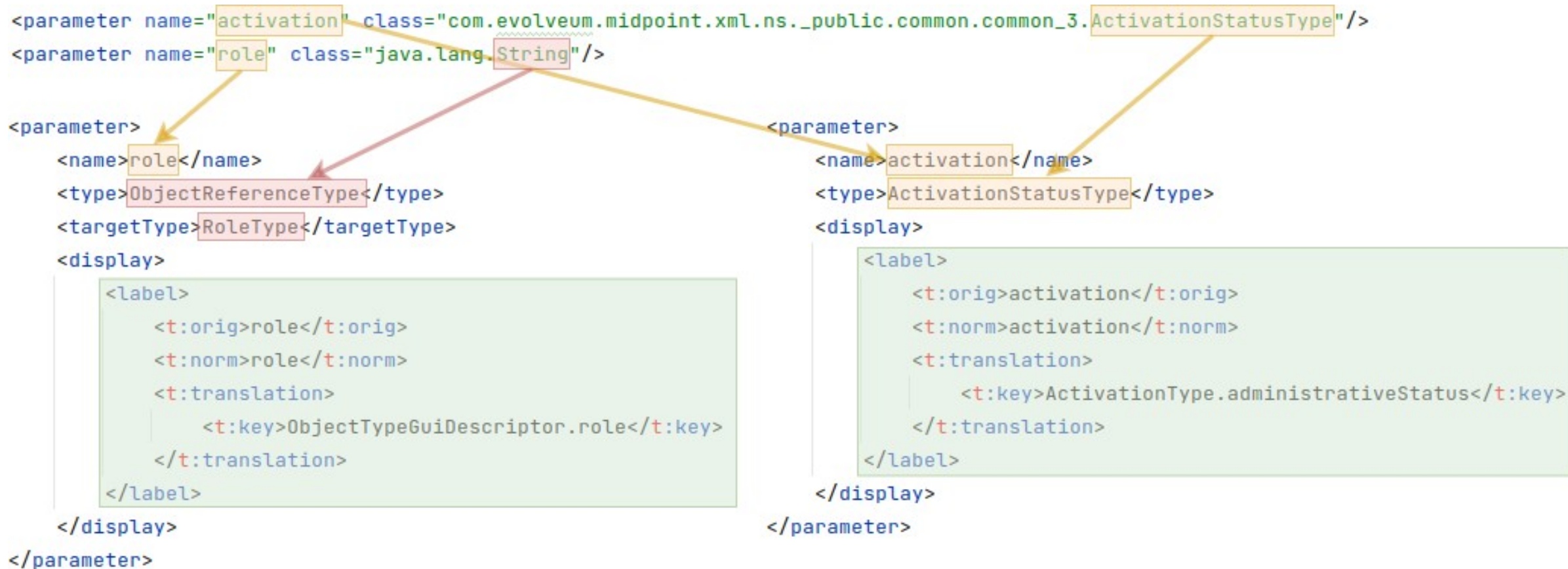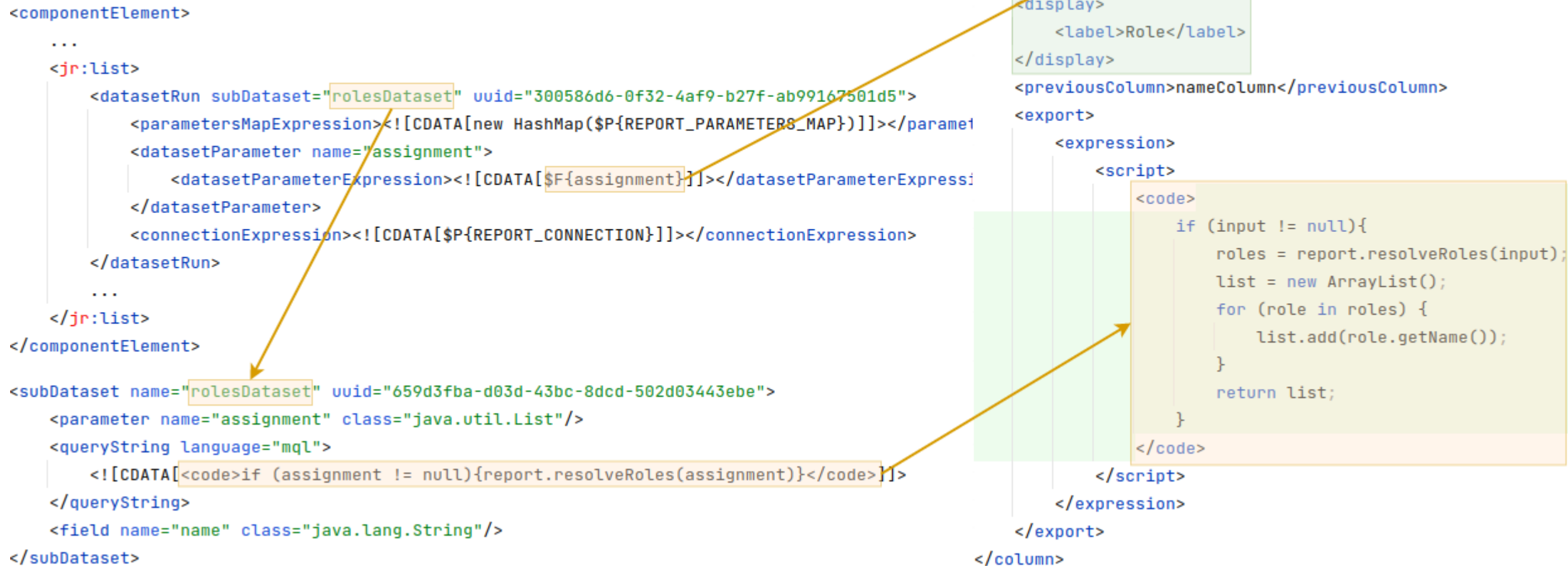
```
<column>
    <name>roleColumn</name>
    <path>assignment</path>
    <display>
        <label>Role</label>
    </display>
    <previousColumn>nameColumn</previousColumn>
    <export>
        <expression>
            <script>
                <code>
                    if (input != null){
                        roles = report.resolveRoles(input);
                        list = new ArrayList();
                        for (role in roles) {
                            list.add(role.getName());
                        }
                        return list;
                    }
                </code>
            </script>
        </expression>
    </export>
</column>
```

Main Takeaways

- Using only Native reports in MidPoint 4.4

- Now we can report more complex use cases

- Try them, and let us know which functionality is missing

- Everything is in the docs!

    - https://docs.evolveum.com/midpoint/reference/misc/reports/configuration/

Thank you for your time!

See more details at https://docs.evolveum.com

Also **follow us** on our social media for further information!

/Evolveum /Evolveum /Evolveum @Evolveum /Evolveum

Evolveum