Workload Automation for Distributed 9.3 and 9.4

Umberto Caselli – Development Manager
Workload Automation v9.3

- May'15: Analyze and Predict Workload Execution
- Dec'15: Expand automation confines with improved and new business integrations (i.e., Hadoop, Netezza, SAP BO, SaleForce, ...)
- March'16: WA Programming Language (WAPL)
- May'16: Simplified deployment and maintenance (central agent upgrade) operations
- Tailored monitoring with customizable dashboard

9.3

- 9.3 FP1
  - Analyze and Predict Workload Execution
  - Expand automation confines with improved and new business integrations (i.e., Hadoop, Netezza, SAP BO, SaleForce, ...)
  - WA Programming Language (WAPL)
  - Simplified deployment and maintenance (central agent upgrade) operations
  - Tailored monitoring with customizable dashboard

- 9.3 FP2
  - Apply changes to what-if experiments
  - Conditional dependencies
  - Import/Export Cron & windows tasks schedules
  - Hadoop Oozie integration
  - 9.3 z
  - Finer-grained security, as a foundation for DevOps
  - Monitoring On-Prem + SaaS workloads from a Single UI
  - Modeling experience re-designed
  - REST APIs allow easy integration with external products and solutions
  - Hadoop Oozie integration (OnPrem)

Intelligent & Hybrid scheduling to implement autonomous decisions across cloud frontiers
v9.4: Collaborative DevOps for Hybrid Workload Automation

✓ Embedded Versioning & Release Management to safely and quickly deploy new applications and changes
  ▪ Version-controlled objects
  ▪ Compare versions
  ▪ Restore previous versions
  ▪ Promotion of changes across lifecycle stages (Dev, QA, Prod)

✓ Integration with ServiceNow for automated tickets initiation

✓ Centralized Auditing and greater accuracy on scheduling changes

✓ Agents upgrade with zero scheduling downtime, enhancement available also for dynamic agents on back level versions!

✓ Reshaped modeling graphical views

✓ Enhanced variable management in jobs and jobstreams

✓ New platform support for agents – CentOS, Ubuntu, SLES Little Endian

*And many other enhancements that increase flexibility and operational efficiencies.*
What if analysis

- Optimize for execution speed with event-triggered workloads and SLA management
- Predict job durations with historical and advanced estimation (cycles, average & variance)
- See the impact of planned and unplanned events beforehand

- Drag & drop jobs in a Gantt view
- Real time simulation of changes on the current plan
- Forecast of impacts
- Trial and error approach

- Detailed statistics are reported for common periods
- Global statistics for all the job instances
- Weekly statistics
- Monthly statistics
- Monthly (from the end of the month) statistics
- Run cycle statistics
Conditional Dependencies

Define workflows with alternative branches based on conditions.

The condition could be the status of the job or any expression that includes the status of the job, the return code, the job log and any job property.
A join is a set of dependencies. It can be used to have more than 40 dependencies (9.4.0.1)

A join is satisfied if \(<n>\) dependencies are satisfied
Agent upgrade

Deploy the patches and run the installer from a centralized location with very limited downtime

Manually from the TDWC  Scheduled with a job
Role based security

The security file is automatically created on the master and on all the backup masters
Java Batch (JSR 352)

Easily integrate your workflows with java jobs on application servers

Define a job with a reference to the java job and pass the parameters
MQTT is a message broker designed for devices

Define a job to publish messages or wait for a message
Hadoop ecosystem

Hadoop is an eco system with many building blocks.

IWS 9.3 is providing adaptors for the Hadoop basic components and for the IBM distribution of Hadoop (Big Insights)

Oozie is a widely used open source scheduler for Hadoop and can create complex workflows in the Hadoop environment.

Integrate IWS and Oozie to provide end to end workflows spanning across different systems

Adding support for Oozie, and to Sqoop and Pig managed by Oozie.
Informatica PowerCenter

Select the “Workflow Details” menu item

A customized view shows the status of each step and let you run action on a specific step
SAP Solution Manager

IWS can be registered as an external scheduler

Define IWS workflows from the SAP UI
SAP Business Objects

Schedule jobs that run reports on SAP Business Objects

Very easy definition of the job
Schedule jobs that run SQL query on a SAP HANA database

**Database Management System**

- **JDBC driver class name:**
  com.sap.db.jdbc.Driver

- **JDBC connection string:**
  jdbc:sap://9.137.35.1176:36015/?autoconn=true

**Credentials**

- **User Name:**
  SYSTEM

- **Password:**
  [Redacted]

  Test Connection
Auditing

**Centralized Auditing (IWSd):**
- Change history
- Justification
- Reporting

**Setting the auditing levels & preferences**

<table>
<thead>
<tr>
<th>Engines</th>
<th>Enable Justification</th>
<th>Category Required</th>
<th>Ticket Required</th>
<th>Description Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>BaromRed - auditing is not enabled</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>CORPORATION_PANTOM_BKM</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>DCC Engine - auditing is enabled only for plan</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>DEMO</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>DEMO Engine - auditing is enabled only for model</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>fancy - auditing is not enabled</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Green - auditing is not enabled</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>no360014</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>no363116</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
</tbody>
</table>
Justification

- The justification is the information about “why” a change has been implemented. It is composed by:
  - the category
  - the description
  - the ticket number
- Set the justification using Dynamic Workload Console or set some environment variables for the command line.
Justification: audit preferences

Use the «default» category or add a new custom category

Define a ticket system in order to use it when the ticket number is required
For all the objects in the model, you can see who made the change, when and why,

In the plan, you can see the history of all the actions performed on an object.

You can see who made the change, when and why.
There is a compare view in the TDWC to show the differences

- A restore button opens an older version in the TDWC
Update a variable table

- A new job type helps in scheduling the update of a variable in a variable table.
- A job can calculate the new value for the variable
Variable Passing: passing variable to the same stream

1. Variable list:
   - Name: ALEV
   - Value: Value

2. Inline script
   - Script body:
     ```
     #!/bin/sh
     .opt/IBM/TWA/TWS/tws_env.sh
     sleep 30
     jobprop VARI valuetext
     ```

3. Variable Table Update
   - Variable Table: ALEV
   - Select variables to update
     - Name: ALEV
     - Value: Value

4. Inline script
   - Script body:
     ```
     #!/bin/sh
     #my business logic
     echo $(ALEVALUE)
     ```

5. General Variables Versions
Open ServiceNow Tickets

This solution helps to open a ticket tracking and reacting at any type of events like:
- Workload Scheduler operations (job in fail, workstation unlinked, job submitted, etc)
- SAP events
- monitoring of the TWS infrastructure
- any custom event defined

Typical scenario: Open a ticket if the job fails
Integrated with the Auditing/Justification feature

Ticket URL

Administrators specify the address of the ticketing server and the specific syntax supported on the server. For example, https://ticket.server.com/ticket/$ticketnumber, where https://ticket.server.com/ticket is the ticketing server sample address while $ticketnumber is the ticket number to be provided by the user.

https://newprod.service-now.com/ticket$ticketnumber

Insert justification

Your administrator requires you to insert justification for changes in the environment.

Category: Trouble ticket

Description: < Trouble ticket text >

OK Cancel
We touch lives.

HCL

TECHNOLOGY • HEALTH • PEOPLE

7 BILLION USD | 105,000 PEOPLE | 31 COUNTRIES

LinkedIn /HCL Enterprise | Twitter /HCL Enterprise | Facebook /HCL Enterprise | www.hcl.com
- Typical DevOps process
- Most of the customers are applying similar techniques
Workload Application Templates

- Promote your definitions from dev to test and production
- Standardize this process
- Rollback in case of problems
- Present since 9.1
- Customers are not yet aware of this functionality

What’s new

- Workload automation templates definition in composer
- New command line to export a template
- Event rules are added to the application templates
Workload Application Templates

- List the job streams belonging to this template
- All definitions referenced by the job streams (jobs, run cycles, event rules, variables) are included
- Import the application template mapping values of dev with values of production.
Workload Automation Templates in composer

The new keyword for workload automation templates is “wat”. A workload application template is a set of job streams.

Promote from your test environment to your production environment, the business applications you develop. Export WAT is now an available option from CLI (wappman).

BAPPLICATION WAT_NAME1
DESCRIPTION "Description"
VENDOR "Provider"
JSTREAMS
AGENT#JS1
AGENT#JS2
END

Usage wappman [connection_parameters]
   -u | -v
   [-import | -replace] <definition_xml_file> <mapping_properties_file>
   [-list]
   [-delete <workload_application_name>]
   [-export <workload_application_name> [-path <export_path>] [-zip <export_zip_filename>]]
   [-display <workload_application_name>]

where valid values for [connection_parameters] are:
   [-file <custom_properties_file>]
   [-host <hostname>]
   [-port <port_number>]
   [-protocol [http | https]]
   [-username <user_name>]
   [-password <password>]

Usage wappman [connection_parameters]
   -u | -v
   [-import | -replace] <definition_xml_file> <mapping_properties_file>
   [-list]
   [-delete <workload_application_name>]
   [-export <workload_application_name> [-path <export_path>] [-zip <export_zip_filename>]]
   [-display <workload_application_name>]

where valid values for [connection_parameters] are:
   [-file <custom_properties_file>]
   [-host <hostname>]
   [-port <port_number>]
   [-protocol [http | https]]
   [-username <user_name>]
   [-password <password>]

Usage wappman [connection_parameters]
   -u | -v
   [-import | -replace] <definition_xml_file> <mapping_properties_file>
   [-list]
   [-delete <workload_application_name>]
   [-export <workload_application_name> [-path <export_path>] [-zip <export_zip_filename>]]
   [-display <workload_application_name>]

where valid values for [connection_parameters] are:
   [-file <custom_properties_file>]
   [-host <hostname>]
   [-port <port_number>]
   [-protocol [http | https]]
   [-username <user_name>]
   [-password <password>]

Usage wappman [connection_parameters]
   -u | -v
   [-import | -replace] <definition_xml_file> <mapping_properties_file>
   [-list]
   [-delete <workload_application_name>]
   [-export <workload_application_name> [-path <export_path>] [-zip <export_zip_filename>]]
   [-display <workload_application_name>]

where valid values for [connection_parameters] are:
   [-file <custom_properties_file>]
   [-host <hostname>]
   [-port <port_number>]
   [-protocol [http | https]]
   [-username <user_name>]
   [-password <password>]
Workload Application Template

Now the Workload Application Template contains also event rules informations in the xml and property file, if the jobstream is referenced in the actions side.

```
  <model:EventRule>
    <model:name>$EVENTRULE_CHECKFILES_DEMO$</model:name>
    <model:json>&lt;eventRule name=&quot;$EVENTRULE_CHECKFILES_DEMO$&quot; ruleType=&quot;filter&quot; isDraft=&quot;no&quot;>&gt;
      &lt;timeInterval amount=&quot;1&quot; unit=&quot;hours&quot;&gt;
        &lt;eventCondition name=&quot;fileCrtEvt1&quot; eventProvider=&quot;FileMonitor&quot; eventType=&quot;FilesCreated&quot;&gt;

#Event rule names
#Assign a name to the event rule to be created in the target environment.
#
EVENTRULE_CHECKFILES_DEMO=CHECKFILES_DEMO
#
#-----------------------------------------------
```