Section 3 - Automation Scripts in Maximo

Section 3: Automation Scripts in Maximo (Interview Q&A with Examples)

Q1. What are Automation Scripts in Maximo?

Answer:

Automation Scripts are small pieces of code written directly in Maximo to extend or customize business logic without deploying full Java classes. They were introduced in Maximo 7.5 and are widely used in 7.6+.

They support multiple scripting languages (Jython, JavaScript, etc.), but Jython is the most commonly used.

They reduce dependency on Java customization and make upgrades easier.

Example Scenario:

In my project, we wrote an automation script to auto-populate the Asset Location on a Work Order when the Asset field was selected. This avoided manual errors and saved time.

Q2. What are the types of Automation Scripts available in Maximo?

Answer:

Automation Scripts can be triggered in different ways:

1. Object Launch Point Runs when events happen on a specific object (insert, update, delete, etc.).

2. Attribute Launch Point Runs when a specific attribute changes.

3. Action Launch Point Runs when triggered by an Action (configured in Escalation/Workflow).

4. Custom Condition Script acts as a condition (returns True/False) for Workflow or Security.

5. Cron Task Launch Point Scheduled jobs (e.g., auto-close old work orders every night).

6. Integration Launch Point (MIF) Runs during inbound/outbound integration.

Q3. Can you explain the difference between Object Launch Point and Attribute Launch Point?

Answer:

Object Launch Point Triggers when operations happen at the object level (e.g., add, update, delete).

Attribute Launch Point Triggers when a specific attribute changes (e.g., LOCATION field changes).

Example:

Object Script When a new Work Order is created, set its Priority = 3.

Attribute Script When STATUS changes to "COMP", set ACTFINISH = Todays Date.

Q4. Write an example Automation Script that sets Work Order Priority = 1 if Work Type = EMERGENCY.

Answer:

Launch Point: Object (on Add)

Language: Jython

```
if mbo.getString("WORKTYPE") == "EMERGENCY":
mbo.setValue("WOPRIORITY", 1)
```

Q5. How do you call an MBO inside an Automation Script?

Answer:

We use the MXServer API to get MboSet.

```
from psdi.server import MXServer
assetSet = MXServer.getMXServer().getMboSet("ASSET", mbo.getUserInfo())
assetSet.setWhere("status='OPERATING'")
assetSet.reset()
asset = assetSet.moveFirst()
if asset is not None:
print asset.getString("ASSETNUM")
assetSet.close()
```

Q6. How can you prevent a user from changing a field using an Automation Script?

Answer:

We can mark the field as read-only by throwing an exception.

```
from psdi.util import MXApplicationException
if mbo.isModified("ASSETNUM"):
```

errorkey = "cannotchange"

params = ["AssetNum"]

raise MXApplicationException("custom", errorkey, params)

Q7. How do you debug Automation Scripts?

Answer:

1. Use service.log("Message") to log messages.

2. Check systemout.log or maximo.log.

3. Use test conditions (if/else) to verify flow.

4. In Maximo 7.6+, you can use the Script Editor with inline testing.

Example:

service.log("WO Status before change: " + mbo.getString("STATUS"))

Q8. Can Automation Scripts interact with workflows?

Answer:

Yes.

Automation Scripts can be triggered from Actions that are part of workflows.

They can also return conditions (True/False) that decide workflow branching.

Example Scenario:

I implemented a script condition to check if a Work Order had open tasks before allowing status change to "COMP". This condition was used in Workflow to block premature closure.

Q9. What is the advantage of Automation Scripts over Java customization?

Answer:

Faster to develop & deploy (no EAR rebuild required).

Easier to maintain during upgrades.

Can be written by functional consultants with scripting knowledge.

Java customization requires more testing and development effort.

But for complex logic or performance-heavy operations, Java is still better.

Q10. What is a Cron Task Automation Script? Give an example.

Answer:

A Cron Task script runs on a schedule.

Example: Auto-close Work Orders older than 30 days.

```
from java.util import Date
from java.util import Calendar
woSet = MXServer.getMXServer().getMboSet("WORKORDER", mbo.getUserInfo())
woSet.setWhere("status='COMP' AND statusdate < (sysdate-30)")
woSet.reset()
wo = woSet.moveFirst()
while wo is not None:
wo.setValue("STATUS", "CLOSE")
wo = woSet.moveNext()
woSet.save()
woSet.close()
```

Q11. How do you pass variables into Automation Scripts?

Answer:

When defining a Launch Point, you can pass input variables (e.g., domain value, attribute). These can be accessed inside the script using their variable names.

Example:

If you pass variable userRole inside script:

```
if userRole == "ADMIN":
mbo.setValue("WOPRIORITY", 1)
```

Q12. Have you worked on any real-time Automation Script scenarios?

Answer:

Yes, in one project we had a requirement to prevent Work Orders from being closed if they had open child tasks. I implemented an Automation Script at the Object Launch Point (on Save) that checked the TASK table for open statuses. If tasks were pending, it raised an error message. This

ensured compliance with business rules.