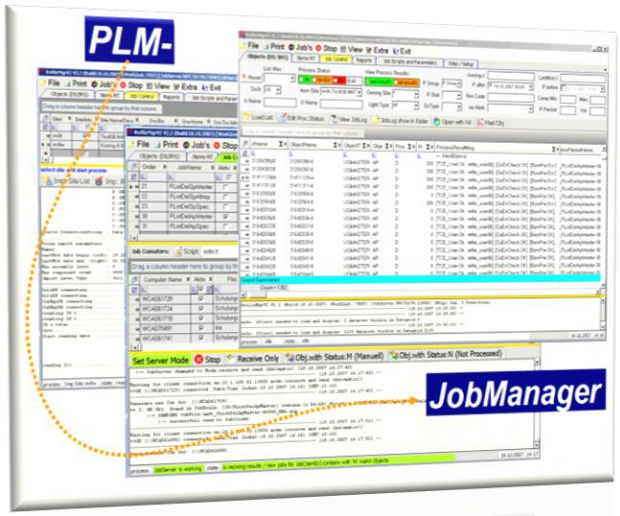


PLMJobManager - CheckBox Validation Report

Validation Report

The following requirements for the validation of the CheckBox application have been tested:

Achieving same results with checkbox and interactive use of NX/TC in following areas:

1. Load and update parts and drawings
2. Data consistency between 2 different NX versions

Additional requirement regarding visualization of checkbox results:

3. Results of CheckBox should be presented in an user-friendly way

1. Test procedure: Load and update parts and drawings

Background:
As descripted on slide 13 **‘CheckBox Data Extraction’** during extracting and performing update actions the program collects issues that appears during this process.
This kind of ISSUES are analysed form logfiles and classified.

Test Procedure:
To validate this requirement you need to open a NX-part, detected by CheckBox, interactively with the same NX/TC Version and settings as the CheckBox extraction process has been used. After opening the part you need to perform the same update actions that has been processed in CheckBox. (Update Feature // Update Drawing) and compare the issues with the issues detected by CheckBox.

Expected Test Results:
Load and update part interactively shall generate the same error message as captured in the CheckBox logfile.

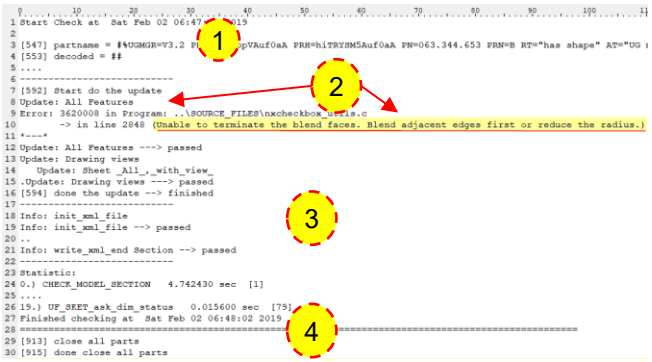
Test evidence is shown in the next 2 slides.

1a. Test results for load and update parts and drawings

CheckBox PLMJobManager

CheckBox

Results from CheckBox Open and Update Process



Process Steps:

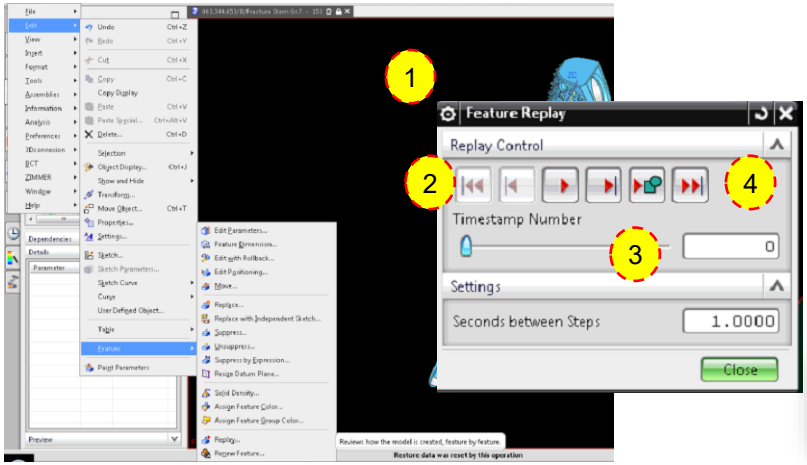
1. CheckBox opens part
2. Do the Update Action and gets message
   Error: Unable to terminate the blend faces.
   Blend adjacent edges first or reduce the radius
3. Extracts CheckBox Data
4. Closes the part
5. JobClient analyses the log file and send result into JobManger Database.

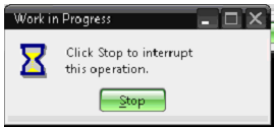
Ir.Id	O.Name	O.P.02 [CBT2(NX1	P.Res.Code	P.Res.CB:Syslog (Eval)	P.Res.CB:UF (update feature) 5
063.344.653/B	063.344.653/B	D.ERR	CB-264	ERR CB (Feature Update Failed)	WRN:Unable to terminate the blend faces. Blend adjacent edges fi...

Both Procedures return the same Messages

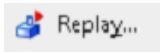
Interaktive Validation

Results from NX Interactive Open and Update Process



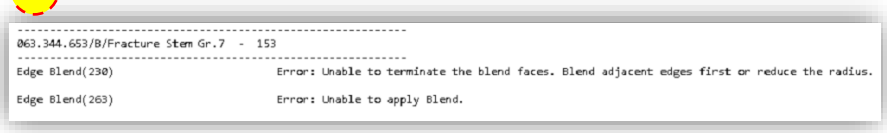


Process Steps:

1. Open Part
2. Start Feature Replay 

3. Start Feature Replay from begin (Timestamp Number = 0)
4. Do Feature Replay until end 
5. After finishing feature replay the Message
   Error: Unable to terminate the blend faces.
   Blend adjacent edges first or reduce the radius.
6. You see in feature tree the feature having ISSUES

5



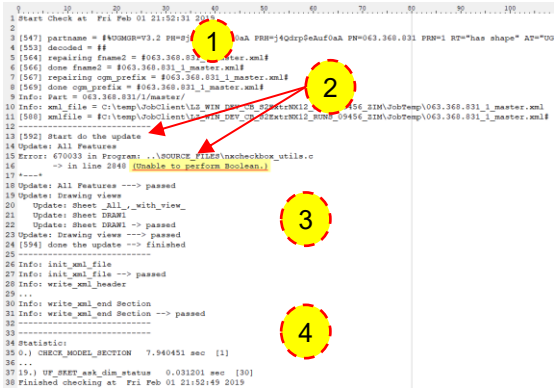
6 

1b. Test results for load and update parts and drawings

CheckBox PLMJobManager

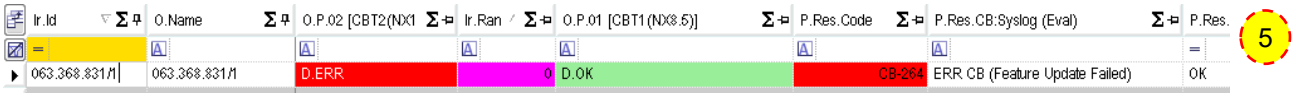
CheckBox

Results from CheckBox Open and Update Process



Process Steps:

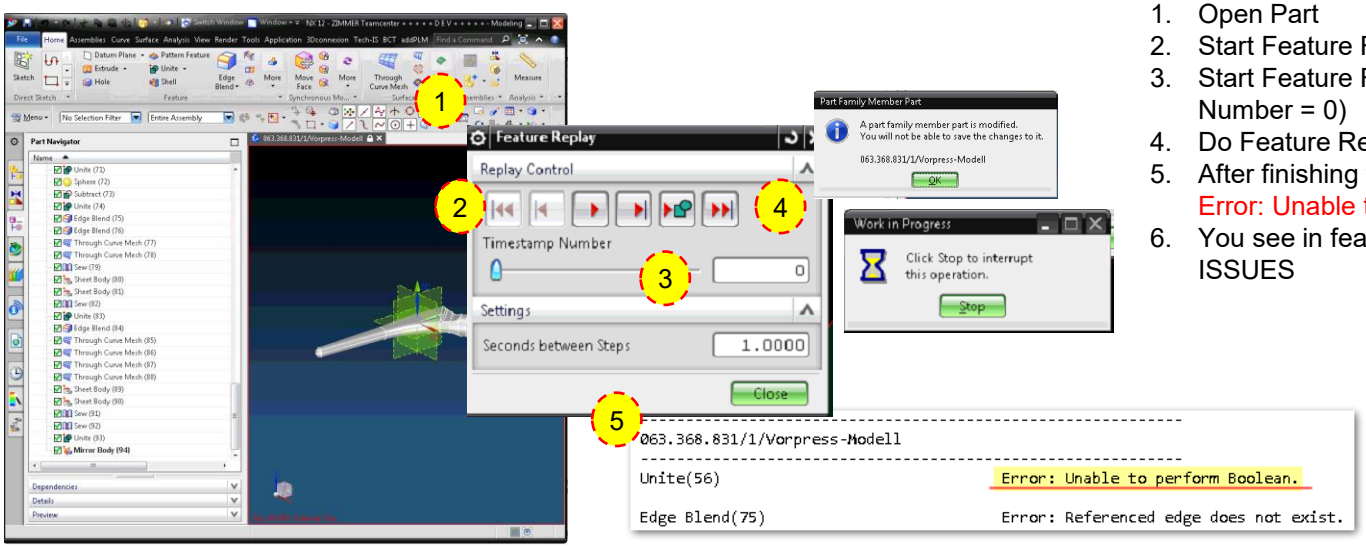
1. CheckBox opens part
2. Do the Update Action and gets message
   Error: Unable to perform Boolean.
3. Extracts CheckBox Data
4. Closes the part
5. JobClient analyses the log file and send result into JobManger Database.



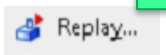
Both Procedures return the same Messages

Interaktive Validation

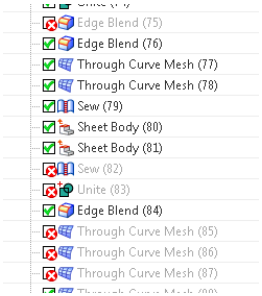
Results from NX Interactive Open and Update Process



Process Steps:

1. Open Part
2. Start Feature Replay 

3. Start Feature Replay from begin (Timestamp Number = 0)
4. Do Feature Replay until end 
5. After finishing feature replay the Message
   Error: Unable to perform Boolean.
6. You see in feature tree the feature having ISSUES



2. Test procedure: Data consistency between 2 different NX Versions

Background:
As descripted on slide 14 **'CheckBox Compare Data'** during comparing CheckBox Data the program collects differences/issues that appears during this process. This kind of differences are analysed by CheckBox compare process and classified.

Test Procedure:
To validate this requirement you need to open a NX-part, detected by CheckBox, interactively with the same NX/TC Version and settings as the CheckBox extraction process has been used. After opening the part you need to do the same update actions that has been processed in CheckBox. (Update Feature // Update Drawing) and compare the output with the results from the CheckBox difference report.

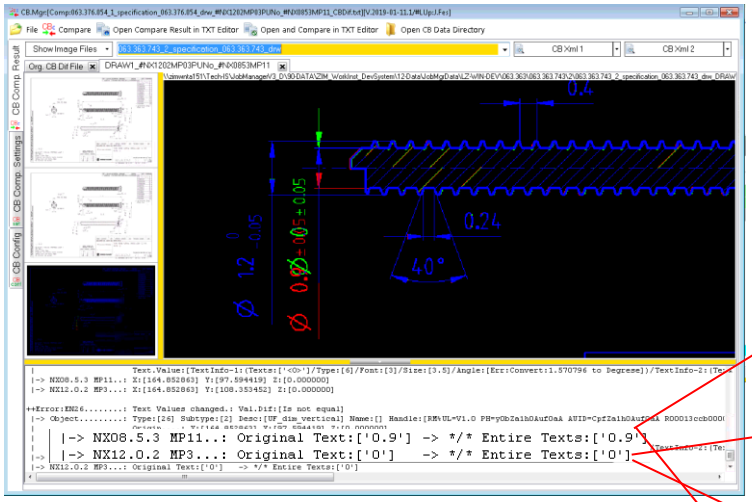
Expected Test Results:
Changes to the drawings by updating it in the higher NX version interactively shall be the same as shown in the CheckBox difference report.

Test evidence is shown in the next 2 slides.

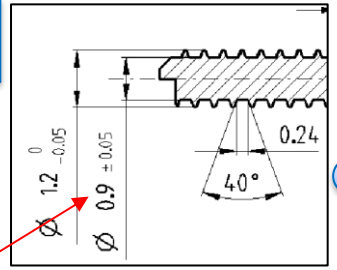
2a. Test results: Data consistency for item #ItemID/RevId#

CheckBox

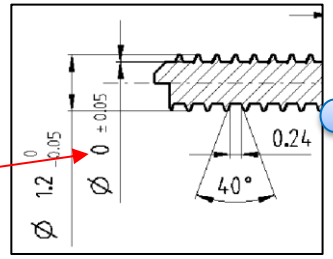
Results from CheckBox Open and Update Process



NX8.5 View



NX12 View

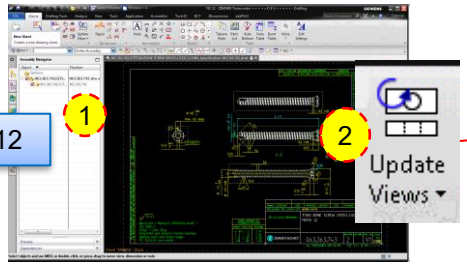


Results
Views updated with same NX version show the same drawing

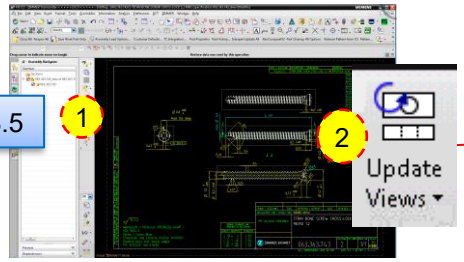
Interaktive Validation

Results from NX Interactive Open and Update Process

NX12

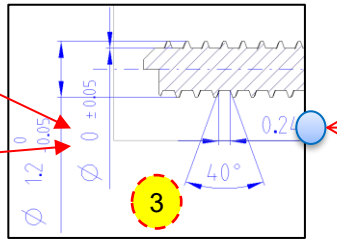


NX8.5

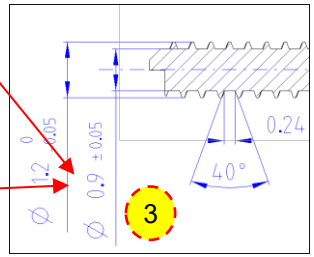


Process Steps:
1. Open Part
2. Start Update Views
3. View updatet Views

NX12 View



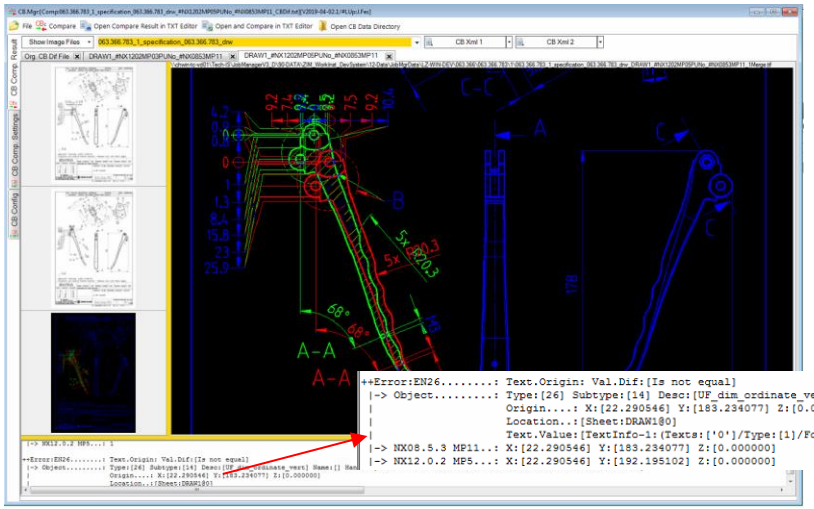
NX8.5 View



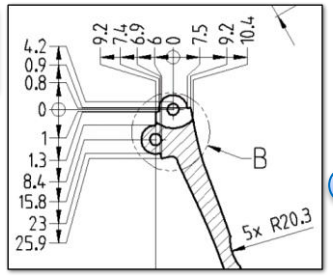
2b. Test results: Data consistency for item #ItemID/RevId#

CheckBox

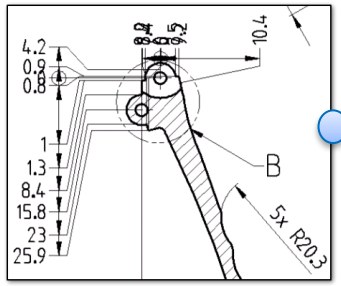
Results from CheckBox Open and Update Process



NX8.5 View



NX12 View

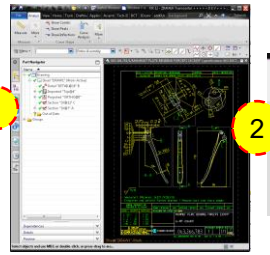


Results
Views updated with same NX version show the same drawing

Interaktive Validation

Results from NX Interactive Open and Update Process

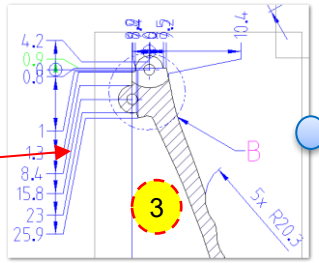
NX12



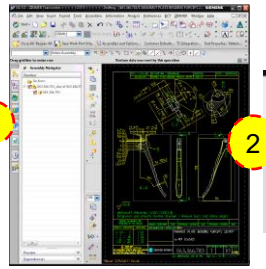
Update Views

Process Steps:
1. Open Part
2. Start Update Views
3. View updatet Views

NX12 View

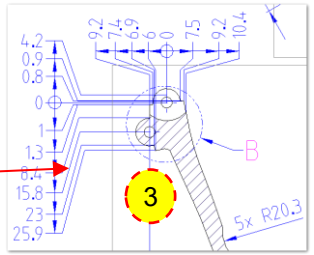


NX8.5



Update Views

NX8.5 View



3. Test procedure: Presenting CheckBox results

Background:
NX Data that having design issues need to be cleaned latest when creating the next revision. Working only within NX the designer does not have the possibility to compare how the drawing looked before and after the upgrade to new software versions.

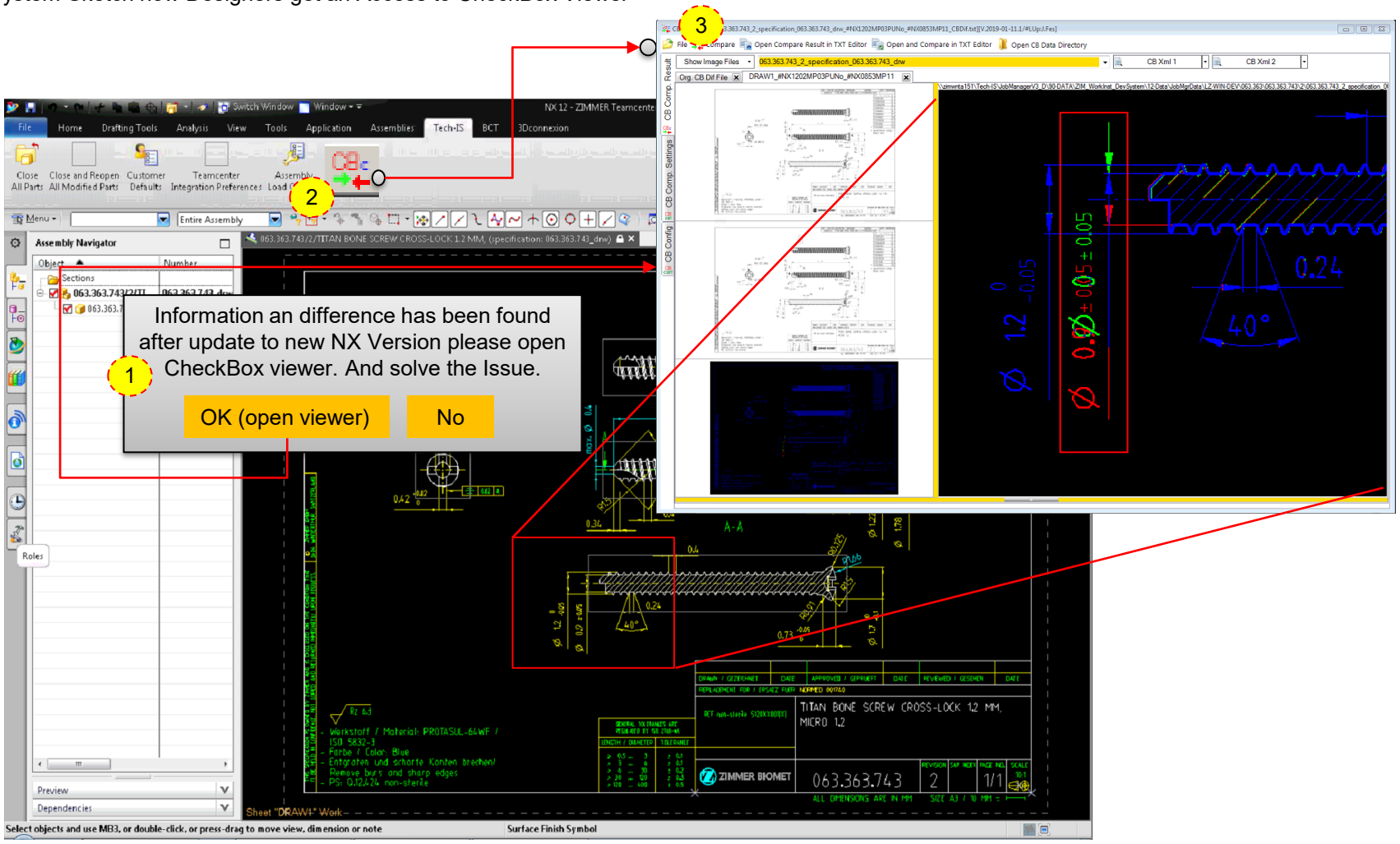
Test procedure:
It shall be possible to compare how the drawing looked before and after the upgrade for the current active work part within the NX application by opening the CheckBox-Viewer.

Expected Test Results:
It is possible to look at the CheckBox results for the current active work part within the NX application by opening the CheckBox-Viewer.

Test evidence is shown in the next slide.

3. Test results: Presenting CheckBox results

System Sketch how Designers get an Access to CheckBox Viewer



Workflow: After Opening a NX part Custom NX.dll checks if an CheckBox diff information exists. User get an Message box **(1)** information. If he clicks on “OK (open viewer)” CheckBox-Viewer opens **(3). Additional he is able to open CheckBox-Viewer via button (2)**

Summary of test results

Achieving same results with checkbox and interactive use of NX/TC in following areas:

Requirement 1:

Load and update parts and drawings

Test results for requirement 1:

NX creates the same messages (Error: Unable to terminate the blend faces. Blend adjacent edges first or reduce the radius and Error: Unable to perform Boolean) interactively as in the CheckBox log file

Requirement 2:

Data consistency between 2 different NX versions

Test results for requirement 2:

The same changes happen to the drawings interactively (moving dimentions) as shown in the CheckBox difference report

Additional requirement regarding visualization of checkbox results:

Requirement 3:

Results of CheckBox should be presented in an user-friendly way

Test results for requirement 3:

CheckBox results can be visualized for the active work part

Conclusion:

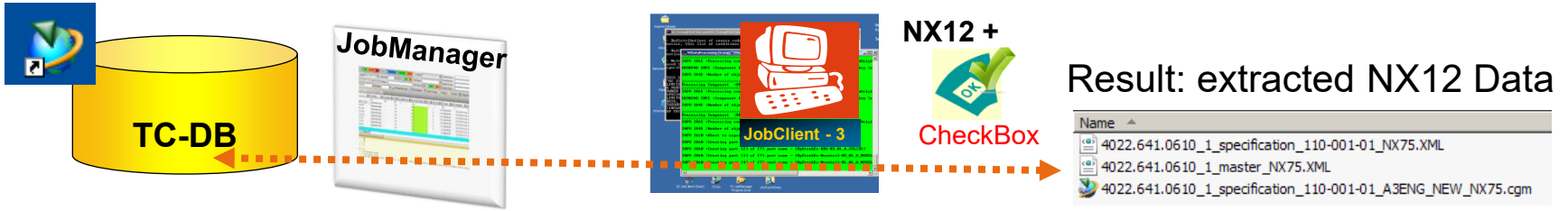
All requirements for the CheckBox application fulfill the acceptance criteria!

Introduction CheckBox Process Overview

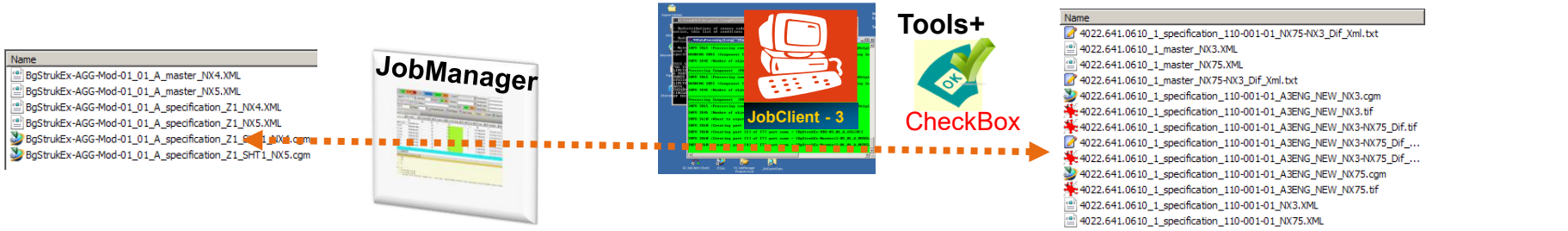
How is CheckBox working?

Step 1: Extraction NX8.5 Data

Step 2: Extraction NX12 Data



Step 3: Generate Analyze Data

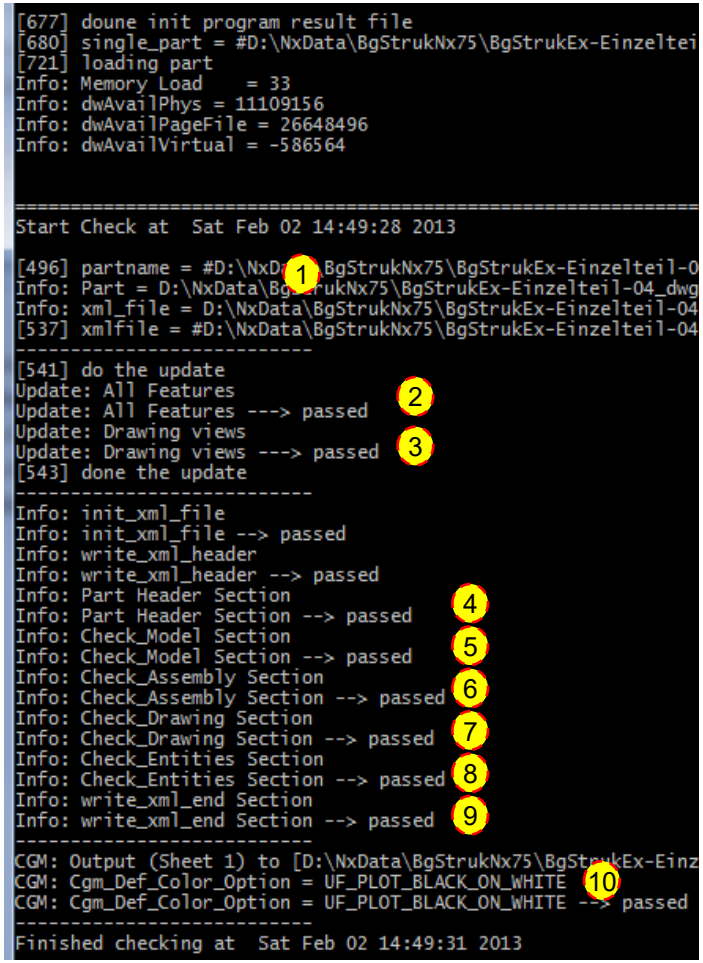


CheckBox Data Extraction

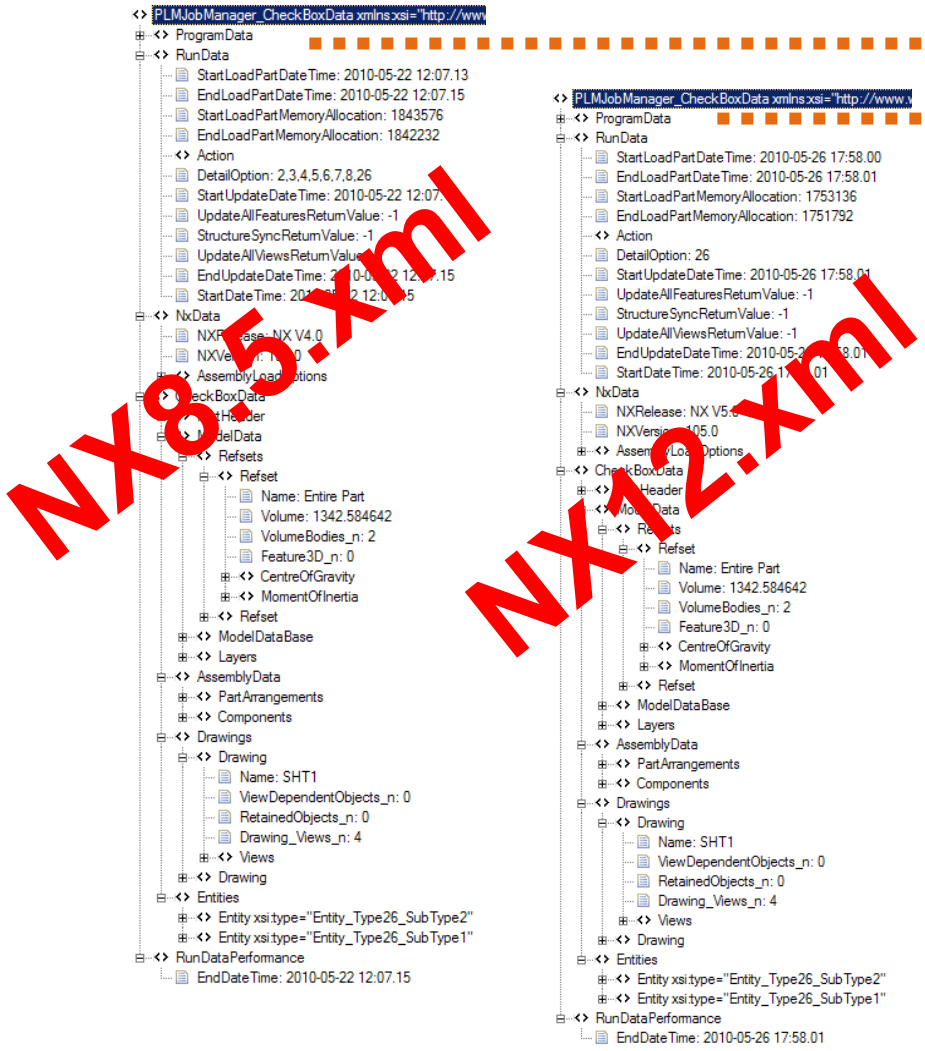
After extracting CheckBox Data the CB.Log files is analysed an the results are listed as partial Results. The following list shows how we do classify the CheckBox extraction Results.

- PL = Part load (1)
- UF = Update all Feature (2)
- UD = Update Drawing (3)
- PH = Part Header (4)
- MD = Model Data (5)
- AS = Assembly Data (6)
- DR = Drawing Data (8)
- EN = Entity (9)
- CBXml = CB.Data File (xml) (10)
- CGM = Drawing .cgm Files

The results of extracting data is imported into the JobServer Database.



CheckBox Compare Data



Compare → having Differences? **YES** **NO**

Create extended Data:
- DifReport.txt
- Dif.tif

In this case it is required to Check what is the reason for this differences !

Result Is OK

Who did used CheckBox for Upgrade Projects.

The CheckBox Software is developed by Mr, Bernd Schieber.
Software specification, project coordination and PLMJobManager integration was done by Mr. Josef Feuerstein (addPLM)
All Company's shown below were using CheckBox for proving legacy data as part of the upgrade process. This Solution is developed since 2008 and in all projects, we did Validation checks.