DX Review Documentation

Version 2, August 7, 2023

# Introduction

## The Purpose of DX Review

DX Review is a lightweight Excel plugin for **Microsoft Excel** that

## Dependencies

This application requires the following libraries, which are often distributed with Microsoft Office products:

* Microsoft XML, v6.0 (msxml6.dll)
* Microsoft Scripting Runtime (scrrun.dll)

XXX

# Quick Start Guide

# Process Flow

There are several steps in the process, some of which must occur outside the control of the program. The first step requires the user to export XML reports from ProjNet (Dr Checks website). It is recommended that full reports, rather than filtered reports, be exported.

# Output Overview

## Regions of Data

Each XML report is summarized into an individual Excel Worksheet.

The data in each Summary sheet can be divided into several regions:

* Metadata
* User Region
* Comments
* Reponses

Each of these regions is discussed in detail below.

## Metadata Region

The root node of the XML summary, called <ProjNet>, usually has two child nodes: the first is <DrChecks> and the second is <Comments>. The <DrChecks> node contains the following project metadata properties, which are expanded into a header in the Summary sheet.

* Project ID
* Project Control Number
* Project Name
* Review ID
* Review Name

It is the final element node <ReviewName> is what is expanded into the name of the Summary sheet. The name of the review is used as the tab name, which does result in lengthy tab lengths. Excel does not permit tab names longer than 31 characters.

## User Region

The following columns are provided in the User Region. This region is located leftmost in a Summary tab to topographically indicate that its data is NOT part of the original XML report content. The following is a list of current fields provided in the User Region

* User Notes
* Action Items
* Assignee

The User Region location and dimensions must not be modified to allow for future enhancements to the plugin.

**Coming Enhancement**

A future enhancement will allow the user to compare values in the User Region between successive versions of the Summary sheet. The user will need to select Summaries to compare. Need to work out what sort of presentation of version control will be implemented.

## Comments Region

As discussed in the **Metadata Region** section, the second child of the <ProjNet> root element node is the <Comments> element, which contains the

As discussed above, the <Comments> node of the XML report contains all the comments, as well as de the *Responses* (a term I am using to include both *evaluations* and *backchecks*). Responses are structured as descendants of a given <comment> ancestor node.

**Developer Note**

When ProjNet exports a report to XML, unfortunately it serializes its data in such a way that the <comment> element is both a descendant and ancestor of itself, which is both unnecessary and bad form. Because of this, to get all the comments correctly the XPATH expression of Comments/comment must be used.

The Comments region is located immediately to the right of the User Region. The comments region does not include all fields provided in the XML. The following properties are included:

* ID
* Status
* Discipline
* Created By
* Created On
* Comment Text
* Attachment
* ***Days Open***

These properties are not in the same order that they appear in the XML <comment> element. There are a few things for the user to know about the data found in several of these fields:

1. The Status is field is either *open* or *closed* and is different from the evaluation field by the same name
2. The foremost condition used to format each row of the Worksheet is determined by the Status field
3. Rather than list the name of the document provided in the Attachment field, a symbol is shown to indicate that an attachment was associated with the comment. Because the attachment is not exported with the XML report from ProjNet, there’s no reason to do more
4. **Days Open** is not a field native to the XML report. It is a calculated field, determined based on the day count from the Created On date

Days Open **Property in Comments Region**

The logic of the Days Open field is as follows: if the comment Status is *open*, the days elapsed between the time when the the code in this program is run and the Created On date is calculated and returned. If the Status is *closed*, the program looks at the Created On date of the last backcheck child of the <comment> and calculates and returns that date difference.

## Response Region

If *evaluations* or *backchecks* are present in the XML report, a Response Region is created to the right of the Comments region.

### Background

In the context of this program, “Responses” is used to define both *evaluations* and *backchecks*. Each <comment> node has <evaluations> and <backchecks> child elements, which have individual *evaluation* and *backcheck* elements.

**Developer Note**

Another issue with ProjNet’s XML serialization approach has to do with the nodeName of individual *evaluation* or *backcheck* element nodes. In an idea XML schema, each evaluation node would be called <evaluation> however, ProjNet appends 1, 2, 3, etc. to each nodeName, e.g., <evaluation1>, <evaluation2>, <evaluation3>. This is both unnecessary and makes naïve XPATH queries cumbersome if they are even possible. Backcheck nodes follow the same unfortunate pattern.

To get around this issue, because each *evaluation* or *backcheck* element are children of the <evaluations> or <backchecks> elements, this program uses XPATH to create an IXMLDOMSelection of <evaluations> and <backchecks> elements, which means the child nodes are all *indexed items*.

How this issue affects naïve traversal of the XML DOM was the major impetus for this major refactor (Version 2).

### Structure of Region

There are two possible ways to order the Response comments: *structurally*, based on the order they appear in the XML tree, or *chronologically*, based on the Created On elements for each comment. In reality, the individually comment children of the <evaluations> and <backchecks> are ordered chronologically. But there are two disadvantages to ordering all Responses based on chronologically.

First, it does require additional, complex code to combine all the comments into a single pool, then order them chronologically. Second, when summarizing the data in the Reponses region of the Excel worksheet, it means that there is no predictability for which column evaluations or backchecks appear in; this could hamper future development.

For these regions, the Response region, which is subdivided into the Evaluations and Backchecks regions, is ordered first structurally (evaluations, then backchecks) then chronologically (for individually comments within the respective evaluation or backcheck regions).

It’s possible to have any number of evaluations and backchecks. When the program initially parses the XML file, it determines the maximum number of evaluations and backcheck comments for all comments in the file. It then structures the Evaluations and Backchecks regions accordingly.

### Presented Data

Similar to the parent <comment> not all the fields present in the XML report are provided in the summary. Here is a list of the data included:

* Status
* Created By
* Created On
* Evaluation Text (or Backcheck Text)
* Attachment

Each backcheck element in the XML does reference the related “sibling” evaluation comment (although these are not technically sibling elements because of how ProjNet serializes the XML file). It was debated whether to include the evaluation IDs, how decided that the context will be clear enough and ultimately can be verified in ProjNet.

The status field has different values from the status field for a <comment>. There are four values:

* Concur
* For Information Only
* Non-concur
* Check and Resolve

The status field of the evaluation comments are used in Excel conditional formatting rules, applied to each comment row then the overall comment status value is *open*.

**Coming Enhancement**

The color coding for open comments is based on the evaluation status parameter. It is possible for there to be more than one open evaluation comment. The current plan is to determine the conditional formatting color based on the highest weighted evaluation status. The proposed weight scale will be *concur* (1) < *for information only* (2) < *nonconcur* (3) < *check and resolve* (4).

This weight order is based on stylistics: *nonconcur*, without much explanation, indicates disagreement, but that can sometimes be resolved by explanation and no further discussion. However, we have used *check and resolve* to specifically mark comments that the AE team wants to discuss during OBR meetings. In that sense, between these two labels, the latter does seem to connote a degree of active coordination that is required to resolve a disagreement.

The user can override and customize the conditional formation, however, to meet their needs.

# Object Model Reference

XXX