A feature that can silently “kills” your Sales order integration

Microsoft D365FO is a cloud product that constantly delivers new functionality to users. Microsoft team is committed to saving backward compatibility, but sometimes you may have regression issues.

Recently there was a big announcement regarding a new Pricing management module (<https://learn.microsoft.com/en-us/dynamics365/supply-chain/pricing-management/pricing-management-overview> ). The module is still in Preview, but let’s consider what happens if you want to try it on your system.

To enable it the new Pricing management the Feature with the same name should be enabled. Then some new functions will be available. Where is the catch: imagine you have some X++ code that creates Sales orders and it has no idea about this new functionality(it may be Integration or some custom functions that creates and post Sales order based on another document)

Typical code for such task is the following

salesTable.insert() .. //Init and create sales header

salesLine.insert() //init and create lines

salesFormLetter.Update //create some document based on Sales order

Pricing management introduces a new code extension that always calls on salesTable.insert() and sets Sales order “Do not process” flag (SalesTable.MrsOrderStopped)

The posting code in salesFormLetter just skips such orders with the “Do not process” flag, and the posting call will be successful without any error or warning. Then you, as a developer, will have a wonderful time understanding why Sales order remained in Open status

Currently, there is no solution for this, the Pricing module is in Preview, and Microsoft promised to fix it in 10.0.37. You can also probably write an X++ code to remove the flag SalesTable.MrsOrderStopped after the SalesTable.insert() call.

If some basic staff stops working (“How to create and post a sales order” is a part of standard X++ dev training), it is very confusing, I hope this article helps you save some debugging time.