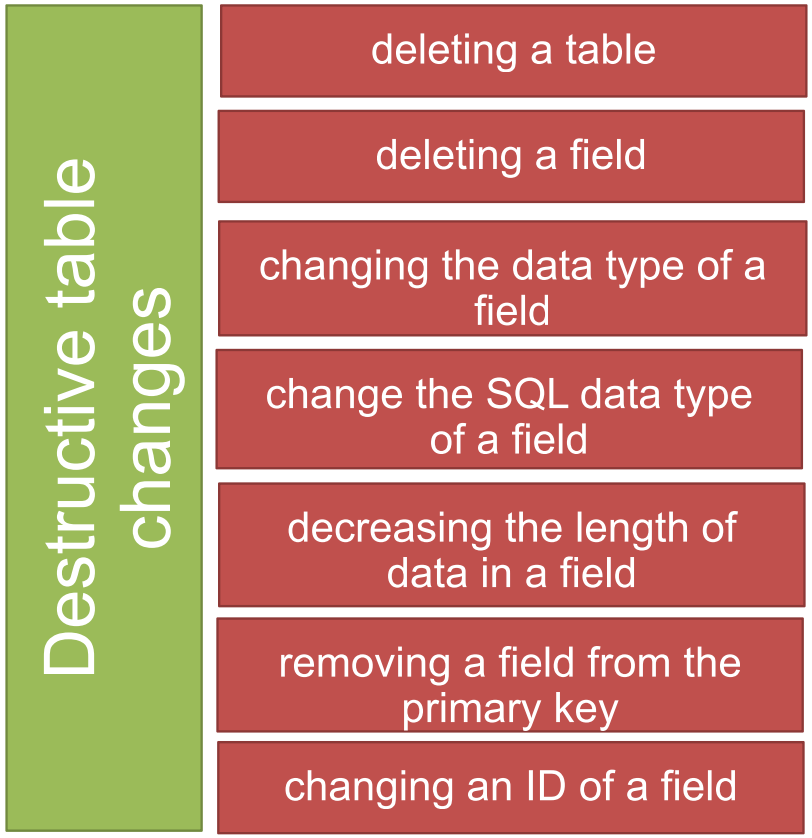
NAV TECH DAYS 2014

mibuso.com

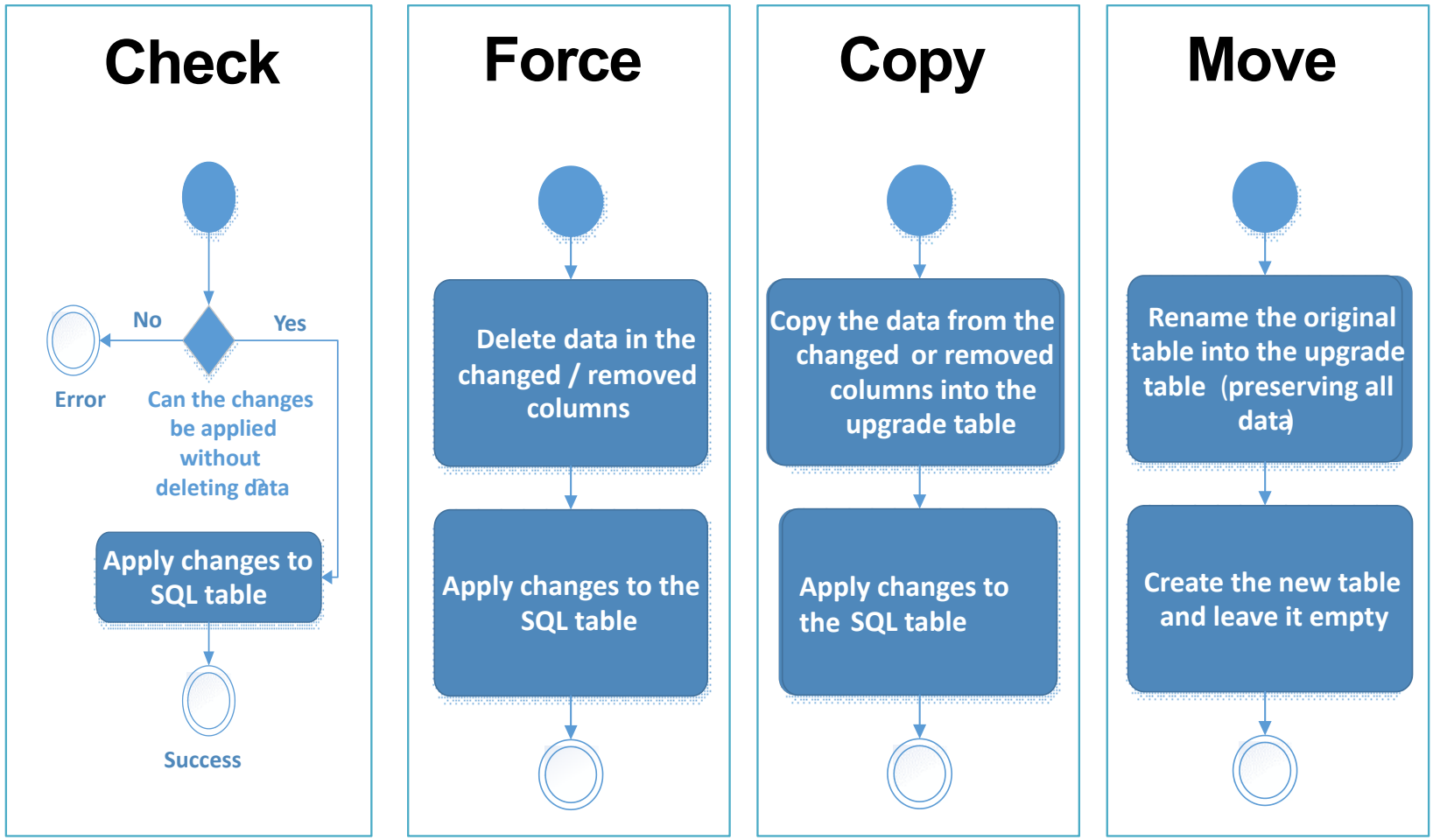
THE NEW PARADIGM OF NAV UPGRADE STORY

Anca-Roxana Ciocan, Bas Graaf, Aida Seifi Labrosse
(Microsoft MDCC)



- Destructive table changes not handled within upgrade instructions are blocked, **regardless** if there is data in the affected columns or not.

- To synchronize a destructive change you can
  - Create or import a specialized **upgrade codeunit with sync. instructions** for how to handle data in this table,
  - **Use the "Force"** synchronization option when saving the change (this action can delete data in the affected columns)


Check
No
Yes
Error
Can the changes be applied without deleting data
Apply changes to SQL table
Success
Force
Delete data in the changed / removed columns
Apply changes to the SQL table
Copy
Copy the data from the changed or removed columns into the upgrade table
Apply changes to the SQL table
Move
Rename the original table into the upgrade table (preserving all data)
Create the new table and leave it empty

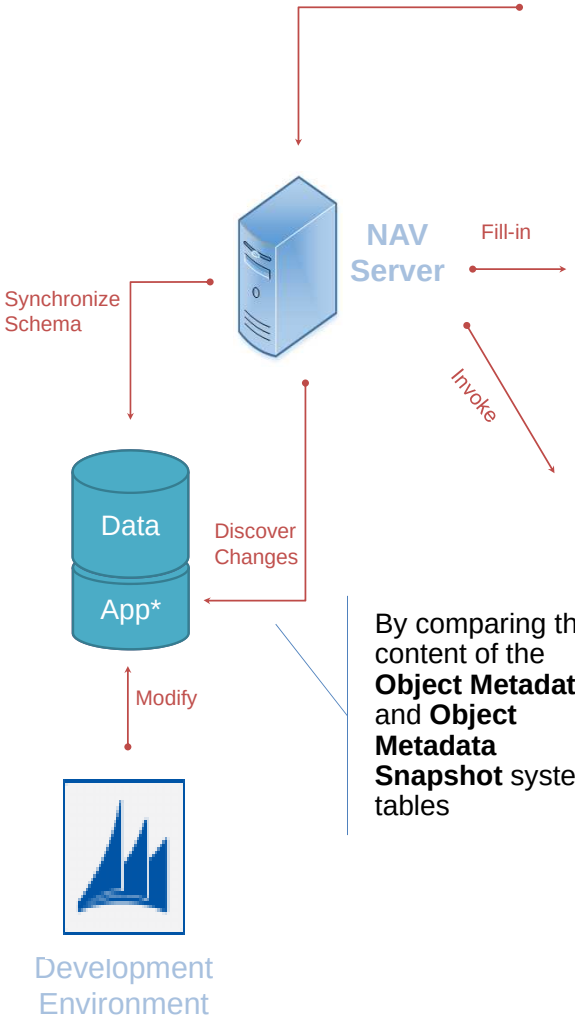
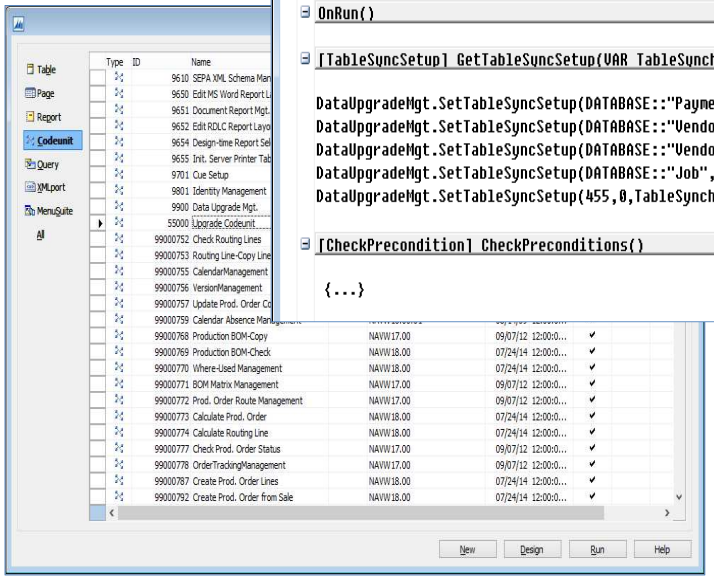


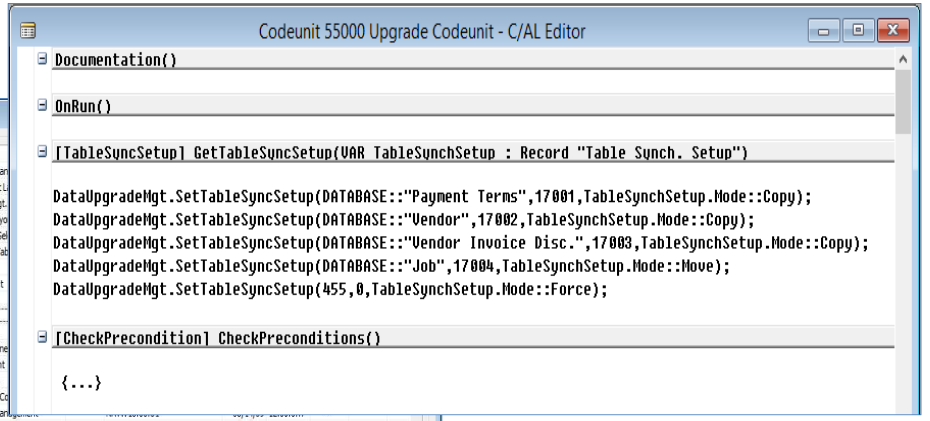
Table ID	Upgrade Table ID	Mode
3	170001	Copy
23	170002	Copy
25	170003	Copy
167	170004	Move
445	0	Force

Fill-in

Virtual Table **2000000135 Table Synch. Setup**, containing IDs of all tables with schema changes

By comparing the content of the **Object Metadata** and **Object Metadata Snapshot** system tables

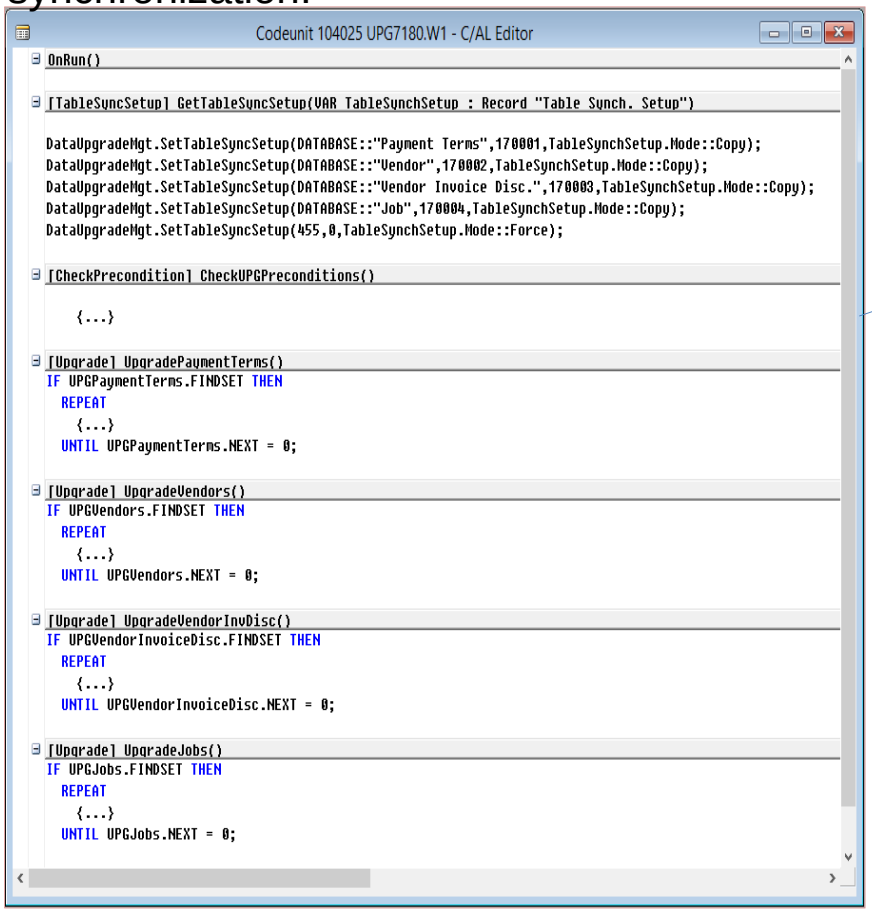




Upgrade Toolkit Step 1 is “outsourced” to the NAV server. The server executes it during the table schema synchronization.

Both steps now reside within a single upgrade codeunit and are **executed by the latest version of the product**.

Upgrade Codeunit



[TableSyncSetup]
Copy or move data from the obsolete/changed tables and fields into upgrade tables or force-delete them if data is not needed.
(previously known as UPGTK Step 1)

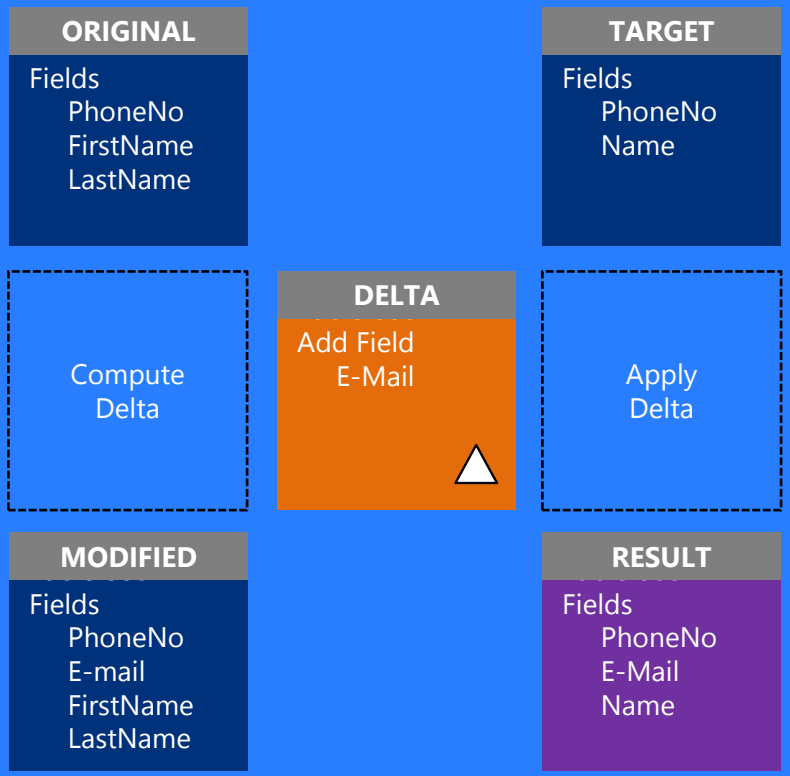
[CheckPrecondition]
Check upgrade preconditions

[Upgrade]
Run custom data upgrade code, process and move the data from the upgrade tables to the new tables.
(previously known as UPGTK Step 2)

3-Way Merge: Object

What?

Detailed example - Deltas



Original

The original unchanged version, source for both of the changes made

Modified

The modified model from which the delta is to be calculated. The source of the delta

Delta

The calculated difference between Original and Modified. (Not always stored)

Target

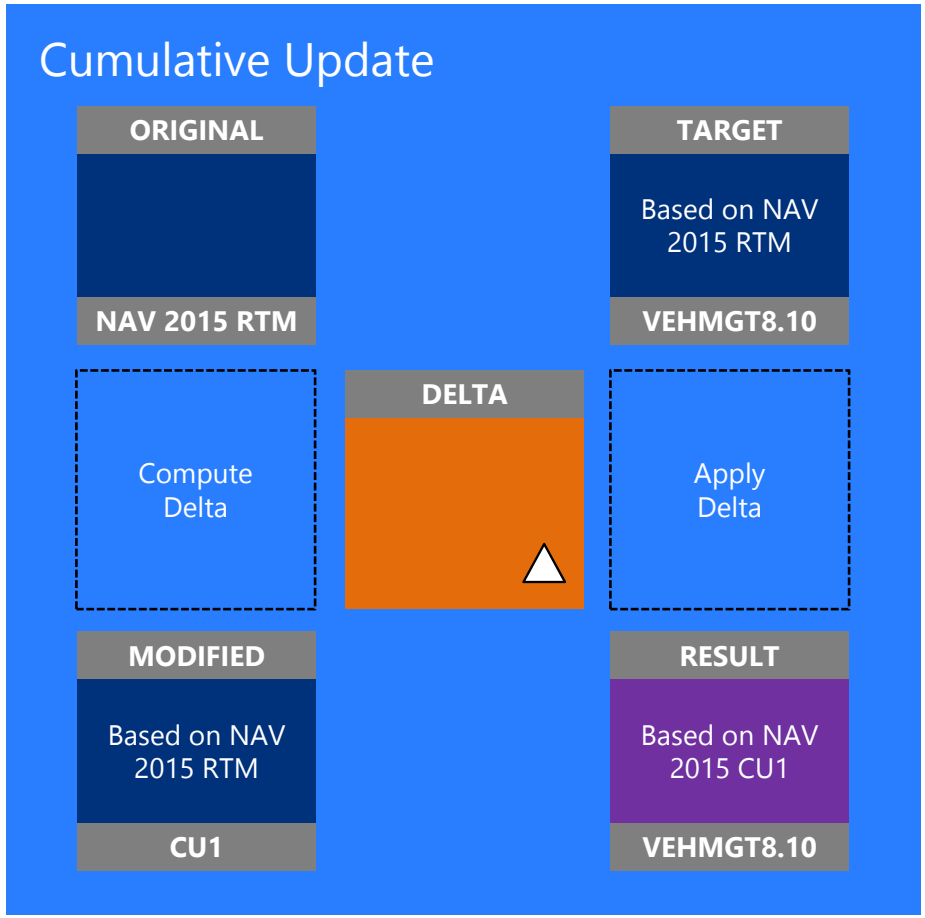
The target model on which the calculated delta is to be applied.

Result

The result of a “Target + Delta” merge operation

3-Way Merge: Cumulative Update

What?



Original

The original unchanged version, source for both of the changes made

Modified

The modified model from which the delta is to be calculated. The source of the delta

Delta

The calculated difference between Original and Modified. (Not always stored)

Target

The target model on which the calculated delta is to be applied.

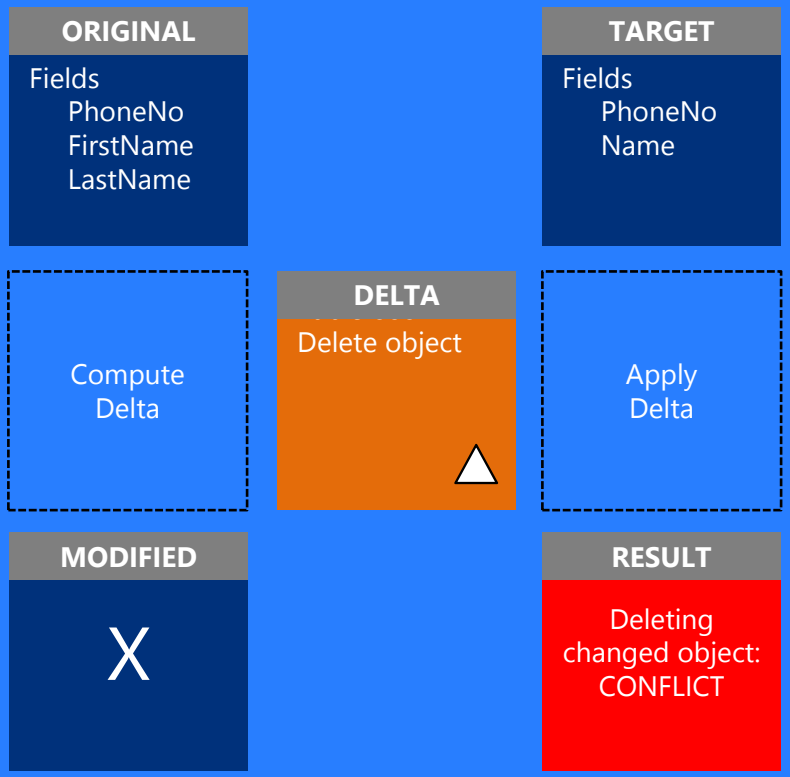
Result

The result of a “Target + Delta” merge operation

3-Way Merge: Object Set

What?

Detailed example – Object Deletion



Original

The original unchanged version, source for both of the changes made

Modified

The modified model from which the delta is to be calculated. The source of the delta

Delta

The calculated difference between Original and Modified. (Not always stored)

Target

The target model on which the calculated delta is to be applied.

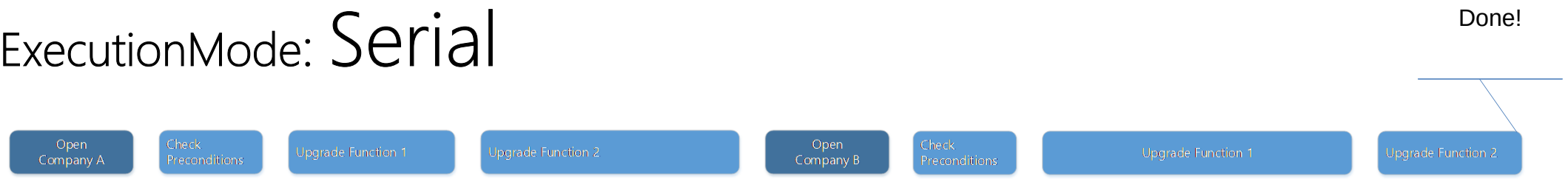
Result

The result of a “Target + Delta” merge operation

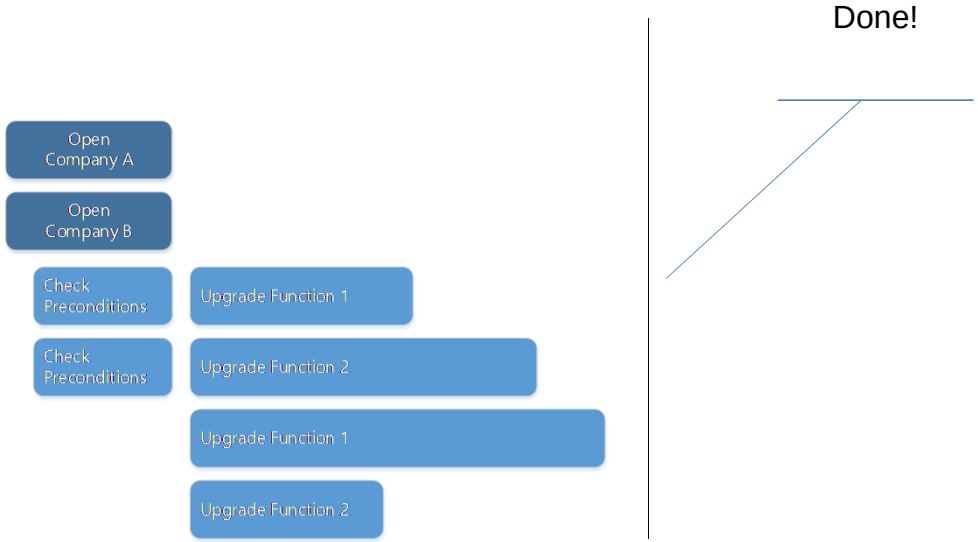
Merge Version List

Original	Modified	Target	Result
NAVW18.00	NAVW18.00	NAVW18.00	NAVW18.00
NAVW18.00	NAVW18.00.00.38457	NAVW18.00	NAVW18.00.00.38457
NAVW18.00	NAVW18.00	VEHMGT8.00 *(custom object)*	NAVW18.00,VEHMGT8.00
NAVW18.00	NAVW18.00	NAVW18.00,VEHMGT8.00 *(customized object)*	NAVW18.00,VEHMGT8.00
NAVW18.00	NAVW18.00.00.38457	NAVW18.00,VEHMGT8.00	NAVW18.00.00.38457,VEHMGT8.00
NAVW18.00	NAVW18.00.00.38457	VEHMGT8.00	NAVW18.00.00.38457,VEHMGT8.00

ExecutionMode: Serial



ExecutionMode: Parallel

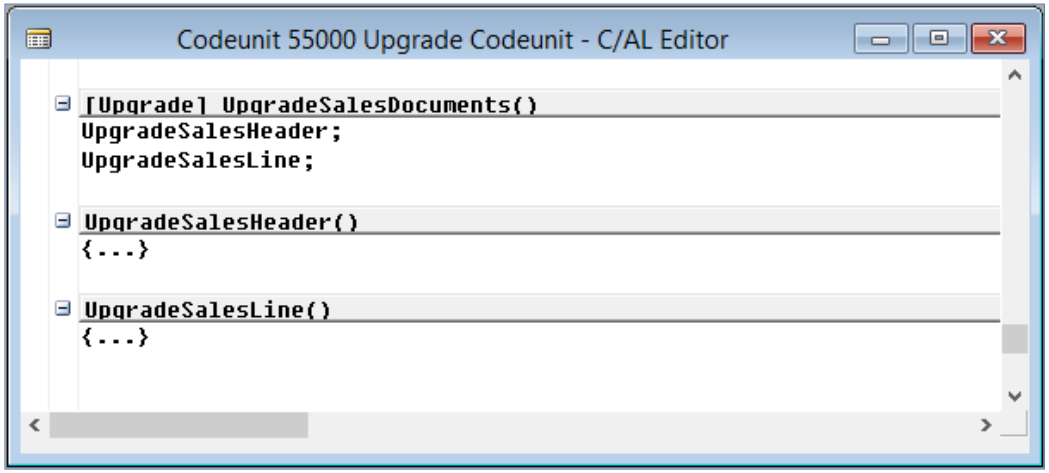


Each upgrade function runs in its **own system session**.

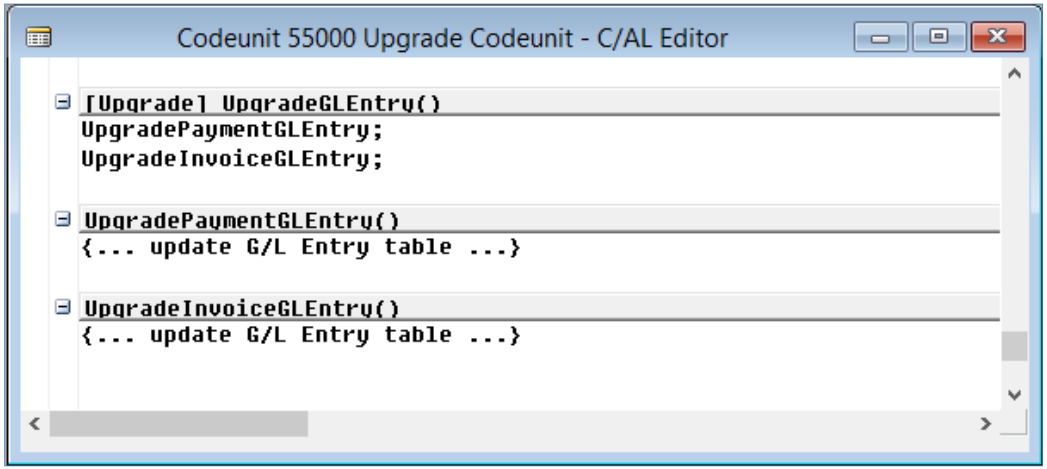
We have **preserved** the ability to run the upgrade functions **sequentially** (for example, in cases where they lock each other).

Data upgrade is an **asynchronous process**, therefore you can start it for **multiple tenants** in parallel too.

Combining order-dependent functions



Combining functions which lock each other



Upgrade functions should be built so that they are:

- Independent of each other
- Not locking each other
- not expecting any particular order of execution

If these conditions cannot be met, you can mark the related (for example, the ones locking each other) and order-dependent functions as [Normal] and then call them from an upgrade function.

Resume

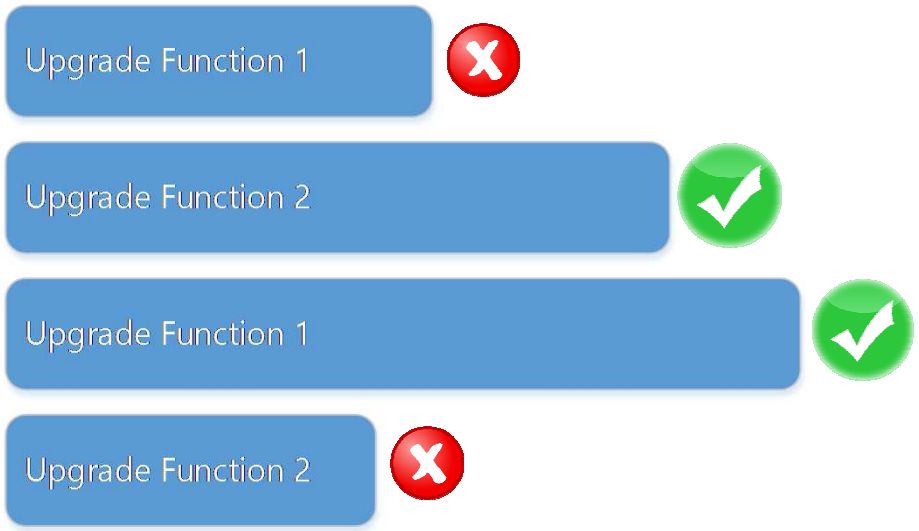
When errors occur in one of the functions:

- the functions which are already completed, will remain completed
- the function which is failed and the functions which were in progress are rolled back and set to a "pending" state

Once the error is fixed, you can resume the upgrade process by restarting the "pending" functions.

You can choose to restart a particular function from a particular codeunit for a particular company.

ContinueOnError



When errors occur in one of the functions , then remaining functions are executed until the process is completed.

.

Benefits

- Simplifies the upgrade environment because all steps are performed by the latest version of the product.
- Increases the upgrade performance because it is possible to execute the data upgrade functions in parallel within the upgrade codeunit and across companies.
- Reduces the amount of code to be written to handle the data upgrade because some actions are now executed automatically by Microsoft Dynamics NAV Server.
- Minimizes the number of manual actions which should be performed during the upgrade, making the upgrade process less error-prone.
- Provides better troubleshooting capabilities collect and address all the errors without rerunning the entire upgrade, Just fix the errors and resume executing what's left
- Uses familiar upgrade toolkit design concepts (such as upgrade tables, upgrade functions).
- End-to-end sample PowerShell script for performing data upgrade available on DVD
- Direct data upgrade path from 2009 (available at cu1), 2013 and 2013 R2 (available on DVD)

Any Questions?

THANK YOU

WHEN YOU ARE PASSIONATE ABOUT MICROSOFT DYNAMICS NAV | *www.navtechdays.com*

NAV TECH DAYS 2014

mibuso.com

LUNCH BREAK

see you back in **60** min.

Data Upgrade

Microsoft Dynamics NAV 2015

- Parallel execution
- Resume after error
- Asynchronous
- Catch all errors
- Detailed progress
- + “old-style” Serial execution and error handling