

iFacto ReVision: User Manual

Datum	09/12/2015
Document Titel	User Manual
Version	2016
Product	iFacto ReVision

Contents

Terminology	4
Repository	4
Source Control	4
SQL Trigger Locking	4
Introduction to iFacto ReVision	5
The Running Clients	5
The Log	5
Client Template	7
Source Control	9
Connect a database/codebase to a repository	9
Step 1: Open the NAV Client, and connect to your database	9
Step 2: Connect client to a repository	9
Step 3: Synchronize All Objects	12
Adding databases by "SQL Servers"	13
Logging code changes	15
Checking the history of an object	16
Use SQL Trigger for extra locking	17
Pending Objects	18
Compare checked out objects with last checked in version	19
Rolling back Objects	20
Hotkeys	22
Other Development Tools	23
Open Client	23
Open Classic Client from ReVision	23
Open RTC Client from ReVision	24
Scan for all NAV Executables	25
Table Information	26
External Editor	28
Compare & Compile	28
Focus windows from Tray Icon	30

Terminology

Repository

A Repository is a database of Sources of your objects. iFacto ReVision supports two repositories: **Microsoft Visual SourceSafe** and **Microsoft Team Foundation Server**.

Source Control

Logging all source-changes to objects in Microsoft Dynamics NAV. Together with this, also prevent over-writing objects by locking objects during development.

SQL Trigger Locking

We use locking in terms of development. So we mean:

- making sure that no one is developing in the same object (object locking) (also in versions before NAV 2009 R2)
- making sure that everything is locked (database locking)
- making sure everyone is mandatory to use ReVision to log changes in the code.

Introduction to iFacto ReVision

iFacto ReVision is a tool that runs in the background. When you start it, you get an icon in the system tray:

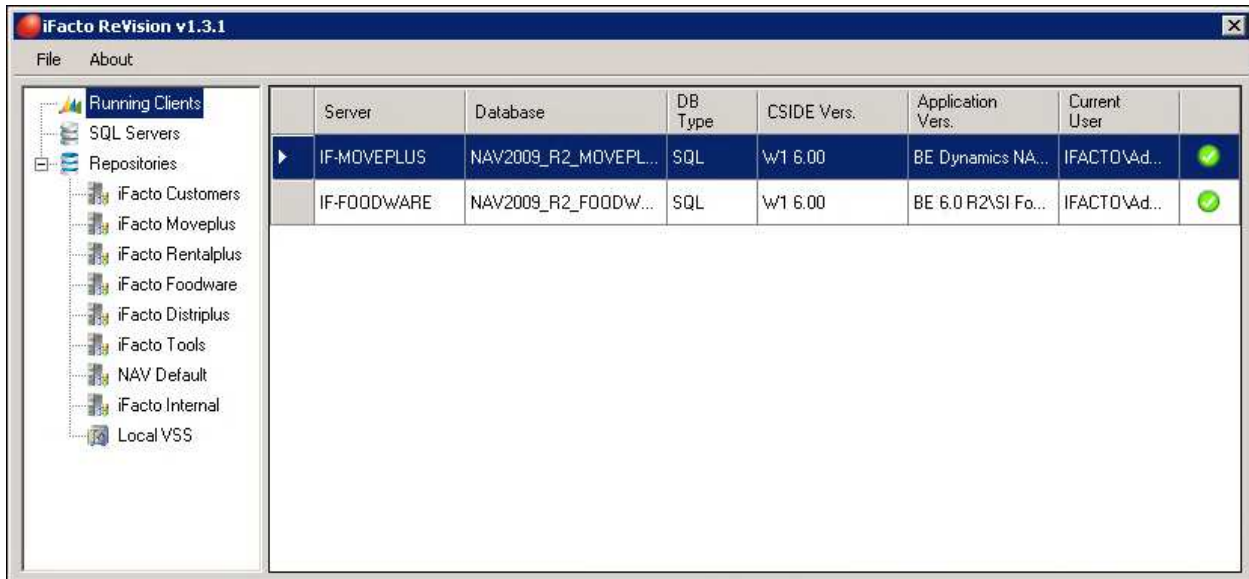


From this icon, a lot is possible, which will be described below.

The Running Clients

When you double click the iFacto ReVision system tray icon, you open the main window.

The “Running Clients” in this window shows you a list of Microsoft Dynamics NAV clients that iFacto ReVision is monitoring. In the following window, I have two NAV clients open, connected to two different databases:



The icon on the right tells you whether it’s connected to a repository or not.

Double click on a client, will take you to either the possibility to add the database, or change its settings.

The Log

From the main window, you can click File / Log. It will show you the logging of the tool:

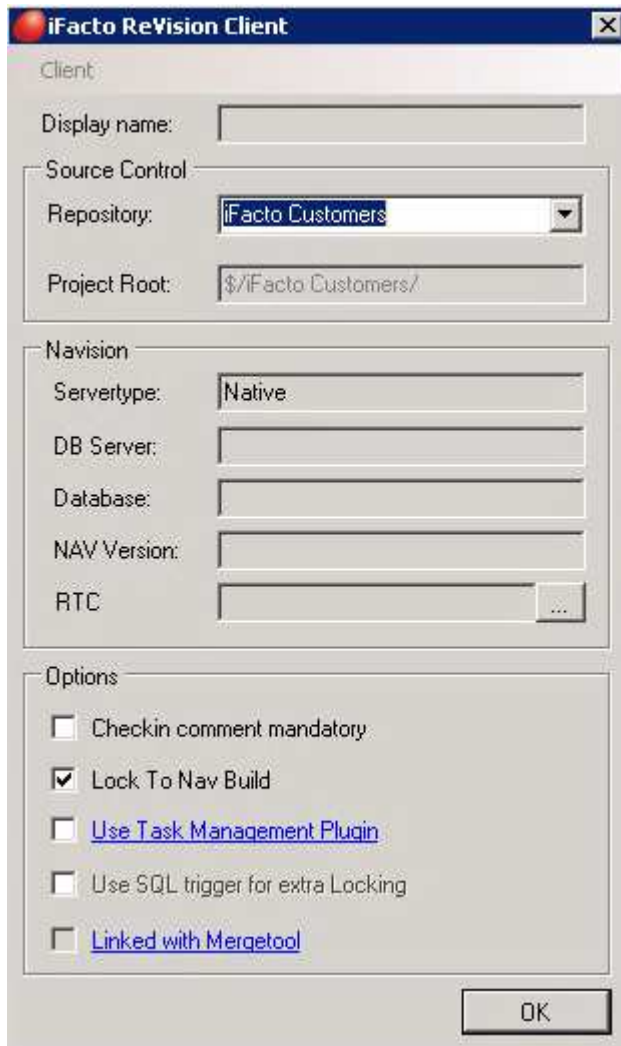
iFacto ReVision - Log			
DateTimeEntry	Session	Module	Description
14/01/2011 9:01:30	iFactoCustomerRep	SourceControl	Auto-confirm mode set to: Automatic
14/01/2011 9:01:30	iFactoCustomerRep	SourceControl	Set workspace
14/01/2011 9:01:29	iFactoCustomerRep	SourceControl	Connect to repository
14/01/2011 9:01:26	iFactoCustomerRep	SourceControl	Auto-confirm mode set to: Automatic
14/01/2011 9:01:26	iFactoCustomerRep	SourceControl	Set workspace
14/01/2011 9:01:26	iFactoCustomerRep	SourceControl	Connect to repository
14/01/2011 9:00:21		Main	Error occured detecting current navision session.
14/01/2011 9:00:19		Main	Error occured detecting current navision session.
14/01/2011 8:46:20		Main	New client detected
14/01/2011 8:45:36	Local Demo DB	NavProxy	Session connected
14/01/2011 8:45:36	Local Demo DB	SourceControl	Set workspace
14/01/2011 8:45:29	Local Demo DB	SourceControl	Load previous checked out objects
14/01/2011 8:45:29	Local Demo DB	SourceControl	Connect to repository
14/01/2011 8:45:29	Local Demo DB	NavProxy	Connect to repository
14/01/2011 8:45:29	Local Demo DB	NavProxy	Connect ROT session
14/01/2011 8:45:29		Licencing	Checking Licence...

This window is there for you to check what ReVision is actually doing in the background. It will tell you when you check out an object, connect to a repository, open a connected client,

You can also go to the Log by right-clicking the Tray Icon or using the hotkey CTRL+ALT+L.

Client Template

From the main window, you can click File / Client Template.



The image shows a screenshot of the 'iFacto ReVision Client' dialog box. It has a title bar with the iFacto logo and the text 'iFacto ReVision Client'. The dialog is divided into several sections: 'Client' with a 'Display name' text box; 'Source Control' with a 'Repository' dropdown menu (currently showing 'iFacto Customers') and a 'Project Root' text box (containing '\$/iFacto Customers/'); 'Navision' with 'Srvtype' (set to 'Native'), 'DB Server', 'Database', 'NAV Version', and 'RTC' (with a browse button '...'); and 'Options' with five checkboxes: 'Checkin comment mandatory' (unchecked), 'Lock To Nav Build' (checked), 'Use Task Management Plugin' (unchecked), 'Use SQL trigger for extra Locking' (unchecked), and 'Linked with Mergetool' (unchecked). An 'OK' button is at the bottom right.

This window is mainly useful for the people who set up the databases in the repository. You can set some default settings, like a default repository (if you have more than one). So, when you want to connect a new database/codebase to a repository, these settings are filled in by default.

Search the repository

You are able to search the repository (or cross-repository) for client connections. Just go to the textbox below, and start typing:

iFacto ReVision v1.3.1 [<DEBUG>]

File About

Running Clients
SQL Servers
Repositories
iFacto Customers
iFacto Moveplus
iFacto Rentalplus
iFacto Foodware
iFacto Distriplus
iFacto Tools
iFacto Internal
NAV Default

Name	Server	Database	NavCler	NavVersion
iFacto DEV	IF-IFACTODB	NAV2009_R2_IFACT...	<input checked="" type="checkbox"/>	6.0.32012
iFacto LIVE	IF-IFACTODB	NAV2009_R2_IFACT...	<input checked="" type="checkbox"/>	6.0.32012
iFacto NAV2013 Trial	IF-NAV2013BETA	NAV2013_BETA_IFA...	<input type="checkbox"/>	7.0.33595
iFacto QA	IF-IFACTODB	NAV2009_R2_IFACT...	<input checked="" type="checkbox"/>	6.0.32012
iFacto Tools DEV	IF-IFACTOTOOLS	NAV2009_R2_IFACT...	<input checked="" type="checkbox"/>	6.0.32012
iFacto Tools NAV2013 Trial	IF-NAV2013BETA	NAV2013_BETA_IFA...	<input type="checkbox"/>	7.0.33595
iFacto Tools QA	IF-IFACTOTOOLS	NAV2009_R2_IFACT...	<input checked="" type="checkbox"/>	6.0.32012
iFacto Tools RLS v1.0	IF-IFACTOTOOLS	NAV2009_R2_IFACT...	<input checked="" type="checkbox"/>	6.0.32012
iFacto Tools RLS v1.1	IF-IFACTOTOOLS	NAV2009_R2_IFACT...	<input checked="" type="checkbox"/>	6.0.32012

\$/iFacto Internal/iFacto DEV/

ReVision will search in the repository you selected. When you select “Repositories”, it will search and show you all links in the repositories below (as you see in the screenshot).

Source Control

When you've set up the repositories (see installation & configuration manual), it doesn't really matter anymore which repository you use, for the developer, it's the same user experience. So we won't make a distinguish here in the type of repository.

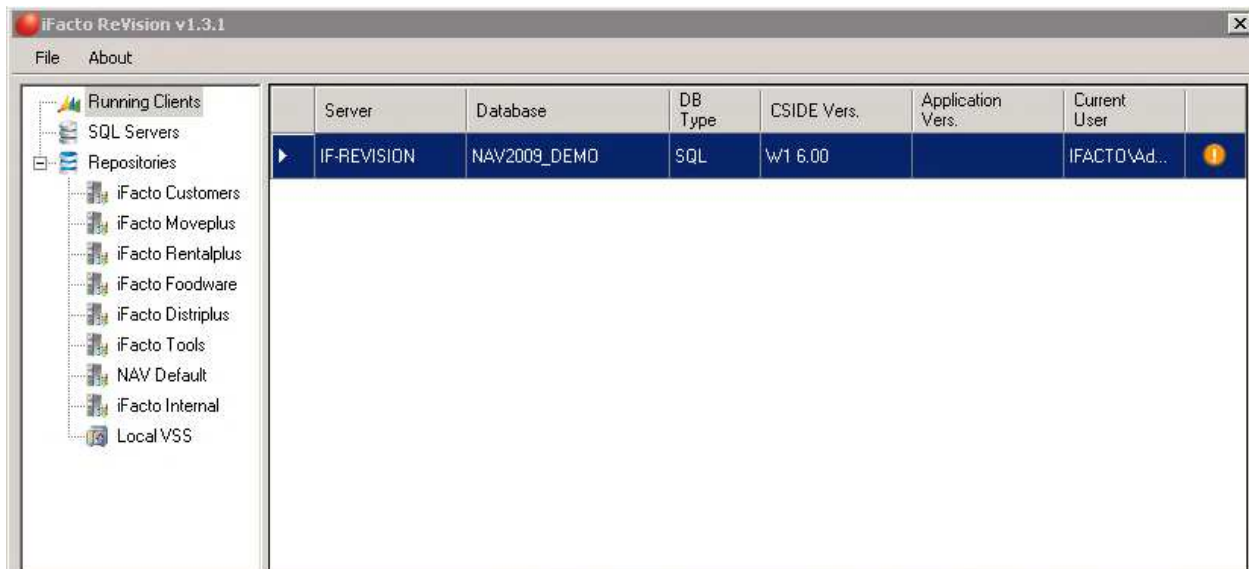
Connect a database/codebase to a repository

Step 1: Open the NAV Client, and connect to your database.

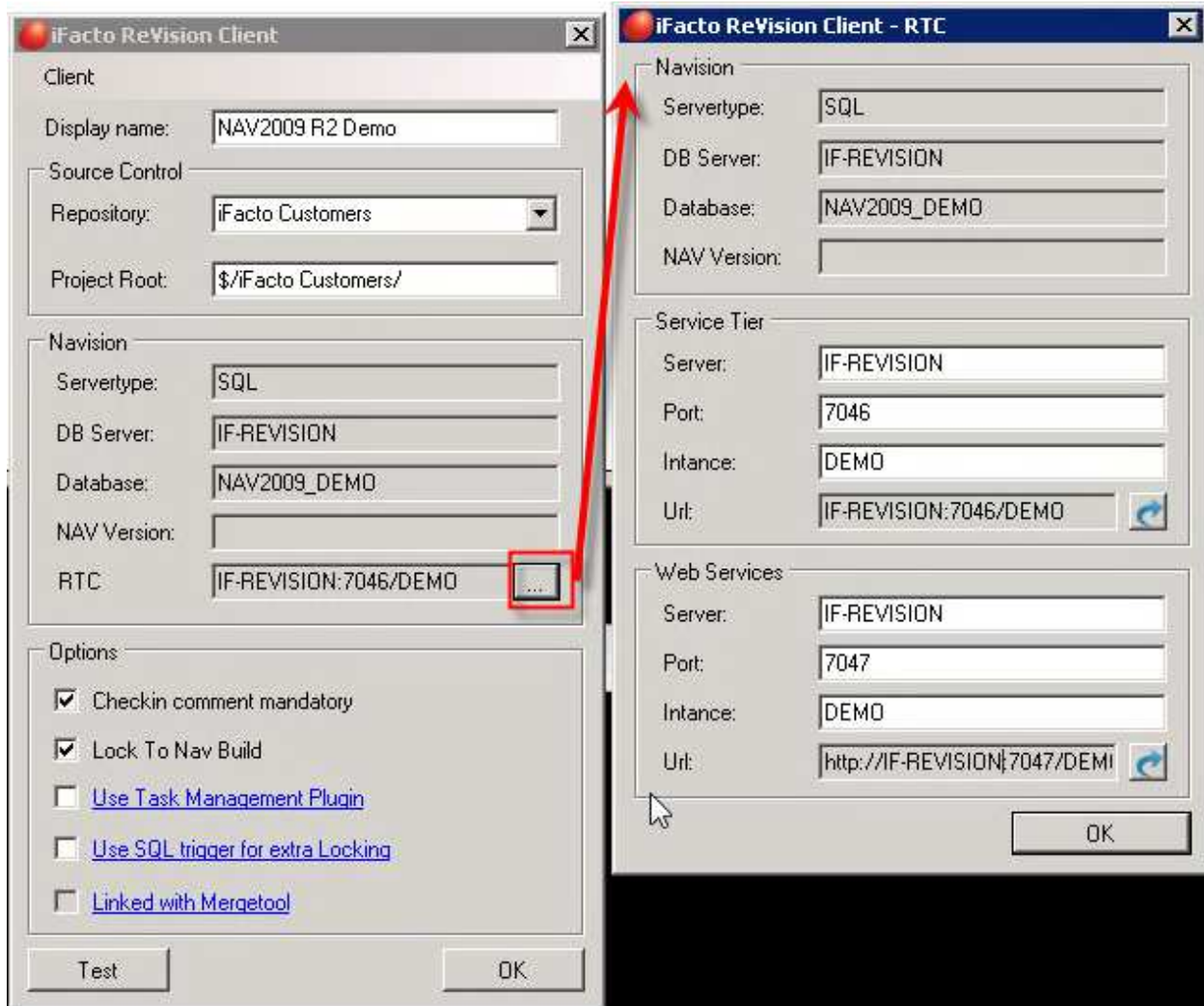
We advise you to always use DNS names instead of using IP addresses. For iFacto ReVision, IP or DNS are two different connections, so two different codebases. Beware of that!

Step 2: Connect client to a repository

In the iFacto ReVision client, you'll see a list of Dynamics Clients that you have opened. Below, only one client is opened:



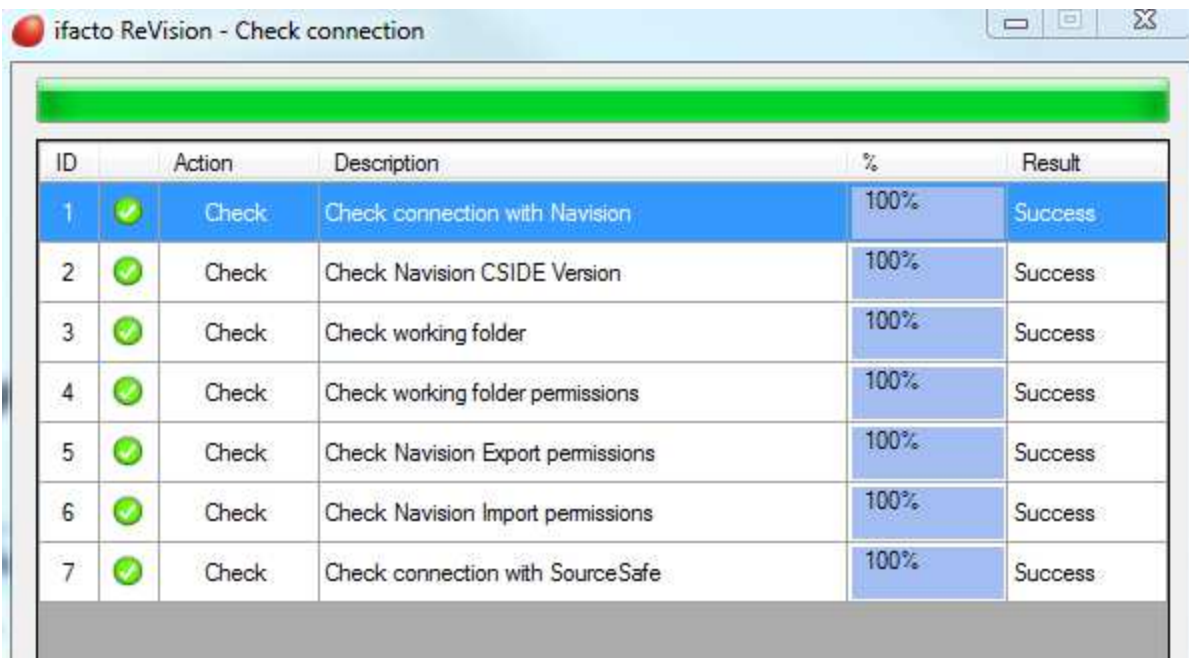
The orange icon indicates that iFacto ReVision couldn't find a connected codebase in any repository. Double click the client and fill in the details.



Field	Description
Display Name	This is the name it will get in your repository. This name is important in your repository strategy. Namely, it IS possible to re-use a name. It will connect to the existing object history, and build further on it. This might be useful for products that jump from one version to another .. (when upgrading or releasing or moving to another database). Tip: never include a database version into your naming convention.
RTC	If needed the location of the service tier can be setup here. This will enabled opening RTC settings to the database from ReVision. Edit the settings by pressing the '...' button next to it.
Repository	Select the repository where you want to log the

	code changes
Use SQL trigger for extra Locking	This is for making sure everyone that connects to this database, will need iFacto ReVision (thus, source logging) to be able to develop in it. This checkbox cannot be enabled directly, choose the 'SQL Lock' option first from the 'Client' menu
Use Project Management Plugin	This option is temporarily not used (later release)
Checkin Comment Manatory	Forces to give a comment when checking in (a set of) objects.
Lock To NAV Build	When checked, ReVision won't allow you to open NAV with another build as last registered by ReVision.
Link With Mergetool	You can connect every database with a central "Mergetool" database. See "Link With Mergetool" at the "Other development tools" section.

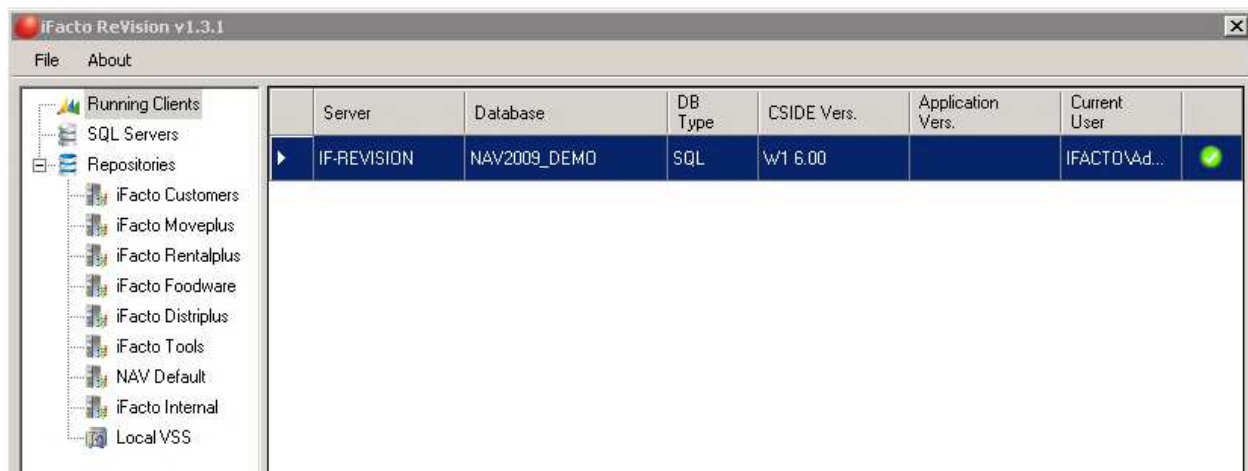
When you click the Test-button, iFacto ReVision will test whether it is going to be able to add this database's sources into the repository:



ID	Action	Description	%	Result
1	Check	Check connection with Navision	100%	Success
2	Check	Check Navision CSIDE Version	100%	Success
3	Check	Check working folder	100%	Success
4	Check	Check working folder permissions	100%	Success
5	Check	Check Navision Export permissions	100%	Success
6	Check	Check Navision Import permissions	100%	Success
7	Check	Check connection with SourceSafe	100%	Success

If successful, close this form, and press OK.

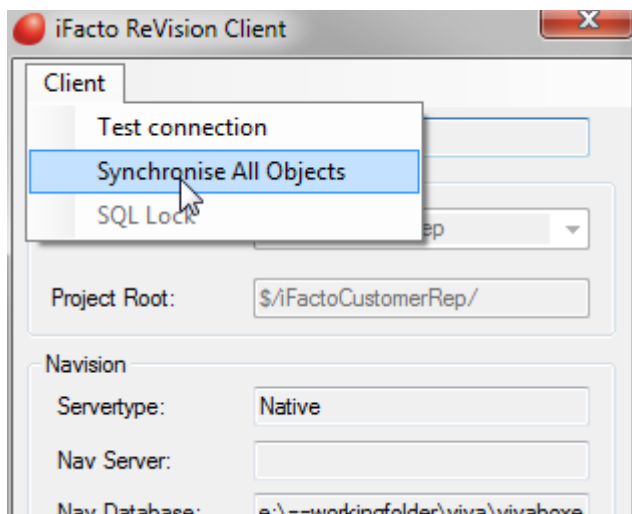
When you click the repository, you see the extra added database/codebase in your repository:



Step 3: Synchronize All Objects

When a connection is set up for the first time, there will be no “base version” of any object in the repository. This isn’t necessary, because iFacto ReVision will automatically create it when designing an object for the first time.

If you do want it, you can just double click the running client, and press Client / Synchronize All Objects.



The iFacto ReVision starts importing all objects in the repository:

ifacto ReVision - Initialize repository

ID	Action	Description	%	Result
1	Check	Check connection with Navision	100%	Success
2	Check	Check connection with Repository	100%	Success
3	Check	Create repository root	100%	Success
4	Check	Create repository subfolders	100%	Success
5	Check	Create working folders	100%	Success
6	UpdateVCS	Create repository subfolders	100%	Success
7	Query	Querying object table	100%	Success
8	UpdateVCS	Update Table in VSS: 940 objects.	6%	Running
9	UpdateVCS	Update Form in VSS: 1762 objects.	0%	NotRun
10	UpdateVCS	Update Report in VSS: 615 objects.	0%	NotRun
11	UpdateVCS	Update Dataport in VSS: 6 objects.	0%	NotRun
12	UpdateVCS	Update Codeunit in VSS: 591 objects.	0%	NotRun
13	UpdateVCS	Update XMLport in VSS: 40 objects.	0%	NotRun

Note: This option is also useful in following example:

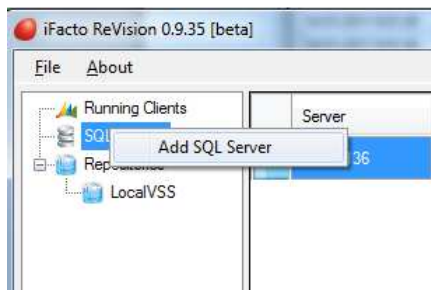
You want to set up a new repository for your product, which has already a number of versions. Just connect to your first version, synch all, remove the link. Connect to your second version, add the database with the same name, sych all, This way, you have one repository with the history of all your product changes..

Adding databases by "SQL Servers"

This option in the Client window is intended to give you an overview of which databases are on a SQL Server, which were not connected with a Repository.

First, Add a SQL Server like this:

- Step 1: Add SQL Server by right clicking the "SQL Server"-entry:

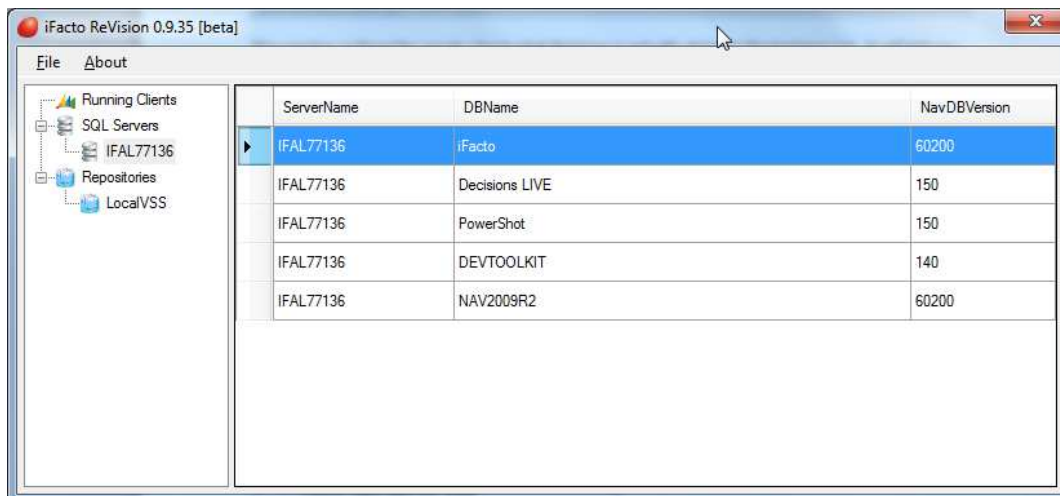


- Step 2: Fill in the Credentials.

We use Windows Authentication. If you don't fill in the credentials, It will use your own windows credentials.



Now, all databases on this server, which are not added to a repository, are shown in a list.



Now, it's possible to add databases to the repository, without even having a client open. Next time you open the client, automagically, it will connect to the repository and people can work with it.

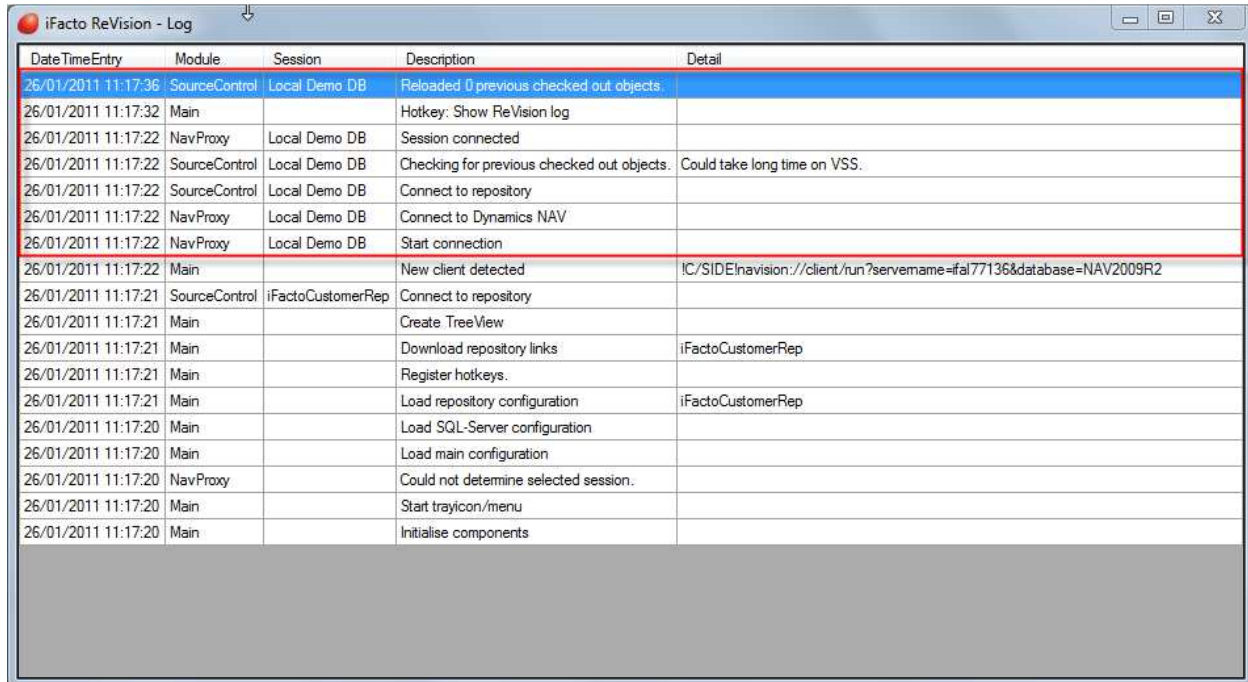
You can add the database by just double clicking, giving it a name, and adding it to one of the repositories.

Note: The "SQL Servers" option is added to make it easy on the the administrators who decide which databases should be logged.

Logging code changes

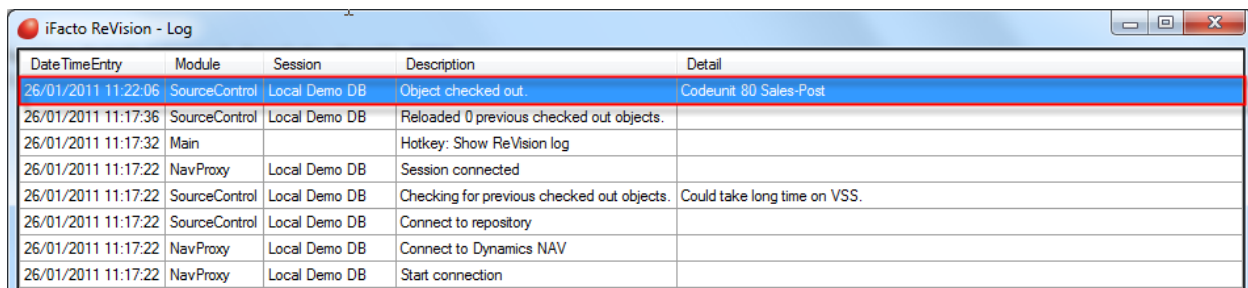
In fact, this is the easiest part. After you are connected, you don't have to do anything else then just developing in your database. Let's work this out with an example.

As you've seen in the above screenshot, I'm connected to my database, so it means that I opened a Dynamics NAV Classic Client, and connected it to a database that is known in my repository. You can also see this in the log:



DateTimeEntry	Module	Session	Description	Detail
26/01/2011 11:17:36	SourceControl	Local Demo DB	Reloaded 0 previous checked out objects	
26/01/2011 11:17:32	Main		Hotkey: Show ReVision log	
26/01/2011 11:17:22	NavProxy	Local Demo DB	Session connected	
26/01/2011 11:17:22	SourceControl	Local Demo DB	Checking for previous checked out objects.	Could take long time on VSS.
26/01/2011 11:17:22	SourceControl	Local Demo DB	Connect to repository	
26/01/2011 11:17:22	NavProxy	Local Demo DB	Connect to Dynamics NAV	
26/01/2011 11:17:22	NavProxy	Local Demo DB	Start connection	
26/01/2011 11:17:22	Main		New client detected	IC/SIDEInvision://client/run?servename=ifal77136&database=NAV2009R2
26/01/2011 11:17:21	SourceControl	iFactoCustomerRep	Connect to repository	
26/01/2011 11:17:21	Main		Create TreeView	
26/01/2011 11:17:21	Main		Download repository links	iFactoCustomerRep
26/01/2011 11:17:21	Main		Register hotkeys.	
26/01/2011 11:17:21	Main		Load repository configuration	iFactoCustomerRep
26/01/2011 11:17:20	Main		Load SQL-Server configuration	
26/01/2011 11:17:20	Main		Load main configuration	
26/01/2011 11:17:20	NavProxy		Could not determine selected session.	
26/01/2011 11:17:20	Main		Start trayicon/menu	
26/01/2011 11:17:20	Main		Initialise components	

When I go into design of e.g. codeunit 80, you'll see nothing happening in NAV (same experience as before), but you do see the following in the log:

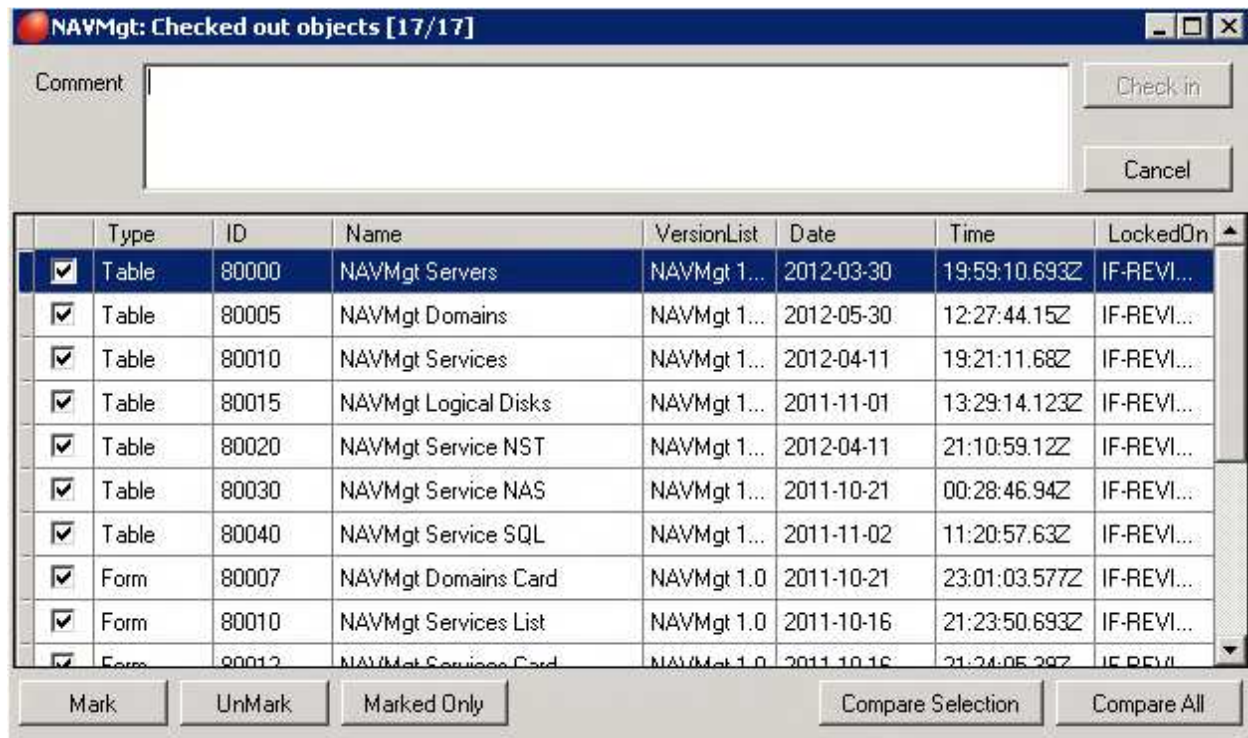


DateTimeEntry	Module	Session	Description	Detail
26/01/2011 11:22:06	SourceControl	Local Demo DB	Object checked out.	Codeunit 80 Sales-Post
26/01/2011 11:17:36	SourceControl	Local Demo DB	Reloaded 0 previous checked out objects.	
26/01/2011 11:17:32	Main		Hotkey: Show ReVision log	
26/01/2011 11:17:22	NavProxy	Local Demo DB	Session connected	
26/01/2011 11:17:22	SourceControl	Local Demo DB	Checking for previous checked out objects.	Could take long time on VSS.
26/01/2011 11:17:22	SourceControl	Local Demo DB	Connect to repository	
26/01/2011 11:17:22	NavProxy	Local Demo DB	Connect to Dynamics NAV	
26/01/2011 11:17:22	NavProxy	Local Demo DB	Start connection	

Codeunit 80 is checked out.

Depending on the setting "Checkin object after form closes", when you close the design of the codeunit, it will check in the codeunit.

When you close the Object Designer, it will show you all pending objects (if any):



In the comments, you can add a project number or whatever, to identify the reason of the change.

You can also check out a number of objects by:

- selecting them in the Object Designer
- Press CTRL+ALT+O or right-click the Tray Icon / Source Control / Check Out

Note: In the near future, the possibility to add reasons to your changes (projects/tasks) will be added to iFacto ReVision.

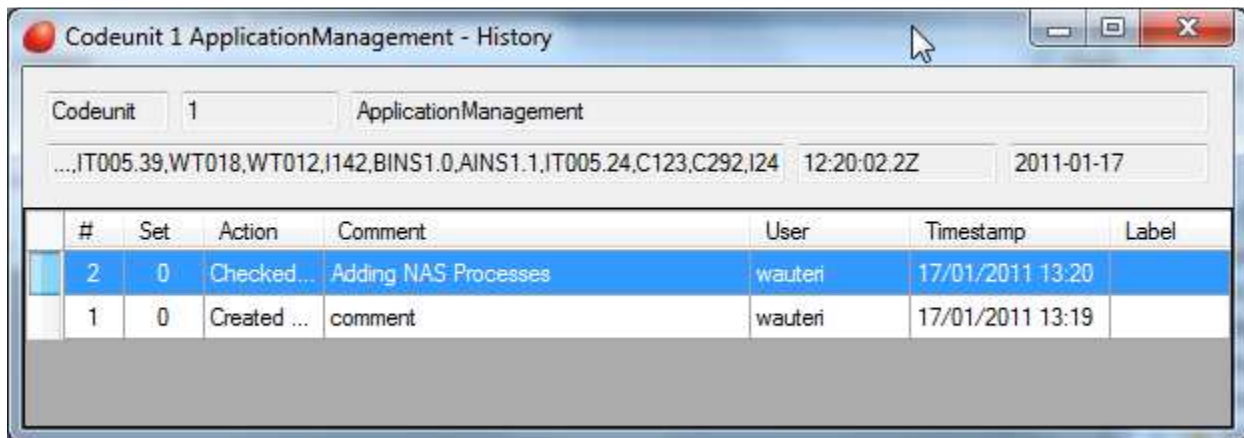
it's important to check in the object from where it was checked out. In the LockedOn column, you can see on which host the object was checked out.

Checking the history of an object

The history of an object shows you all the changes, when and by whom.

Step 1: Fully select the line of the object in the object designer. This is important, else ReVision won't know which line was selected.

Step 2: Right-Click the Tray Icon / Source Control / Show History



As you can see, we have two versions of codeunit 1.

In this window, you can:

- **View** any version of the object in a text editor
- Select either two objects, and **compare** its version by Right-Click / Compare. For this, you need a comparetool configured in the Preference window.
- **Compare** any version of the object **with the current saved version**.
- **Rollback** to any version of the object (see separate paragraph "Rolling Back Objects")
- **Merge on Object:** Merge 2 versions of the history. Saving will save into the DEV environment (when "Live Compile" is enabled).

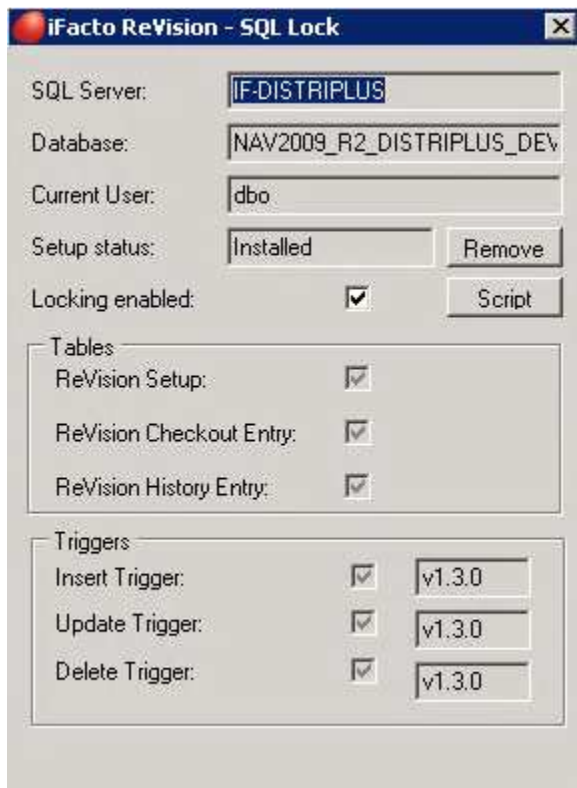
In this compare-functionality, it is NOT possible to save to NAV!

You can also use the hotkey CTRL+ALT+H to check the history of an object.

Use SQL Trigger for extra locking

For making sure everyone uses ReVision (so ever change is locked) we have to introduce a SQL Trigger. The installation of this trigger can be done with the iFacto ReVision client, by a user who has sufficient rights on the database.

Step 1: In your Repository window, double click the database/codebase on which you want this extra locking. Click Client / SQL Lock.



You see the statuses:

- Is the trigger installed?
- Is the trigger enabled.

When exiting the window, you'll see that the check mark is checked. From then on, users are mandatory to install and configure iFacto ReVision before doing changes to the database. Also, the objects are really locked for other developers (for all supported versions of NAV Classic client).

Note: The version of the SQL Lock does not need to correspond with the version of iFacto ReVision. The version of the SQL lock corresponds with the version of ReVision where it was updated for the last time.

Pending Objects

When you start a Client and open the Object Designer, iFacto ReVision will be trying to connect to it. When connected, ReVision will load the already-checked-out-objects-by-yourself. This is shown by a progress bar:

iFacto ReVision - Lenaerts Blommaert NV : Reload pending objects

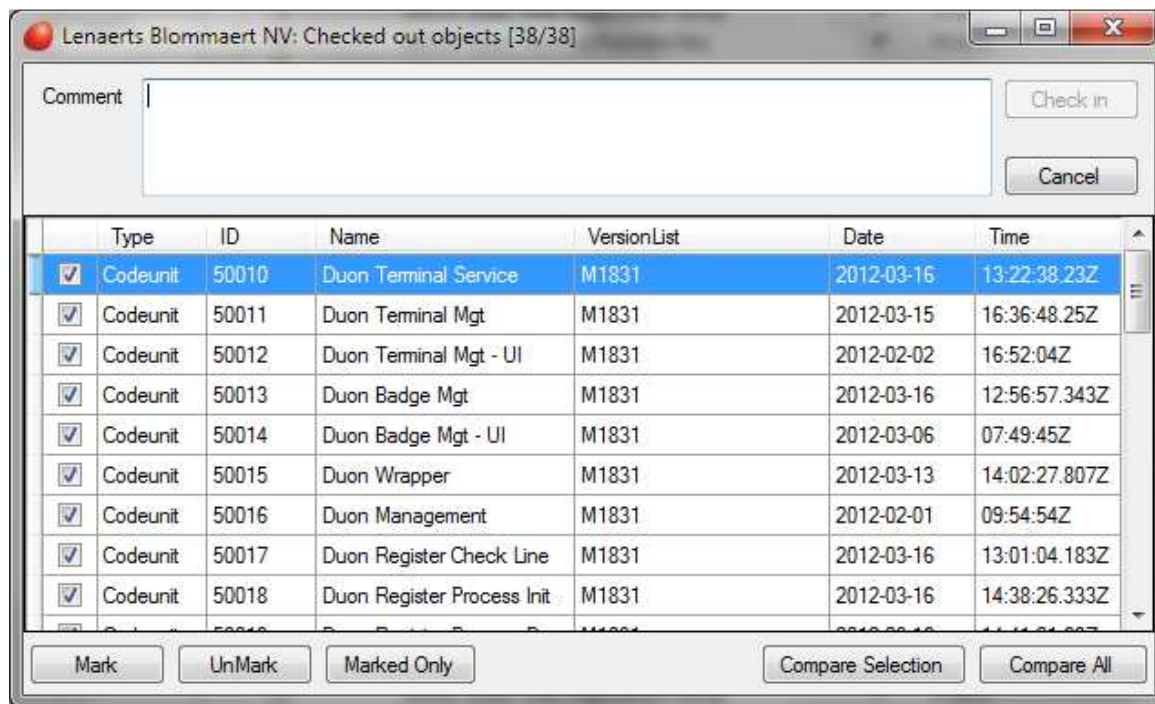
ID	Action	Description	%	Result
1	Query	Reload pending object from repository [38]	100%	Success
2	Checkin	Reload Codeunits: 13 objects.	100%	Success
3	Checkin	Reload Pages: 14 objects.	100%	Success
4	Checkin	Reload Tables: 10 objects.	0%	Running
5	Checkin	Reload XMLports: 1 objects.	0%	NotRun

To show the pending object, just right-click the Tray Icon / Source Control / Pending Objects or use the hotkey: CTRL+ALT+P.

From this window, you can check in all objects or a selection of the objects. You can also compare all the developments you've done with the last checked in version.

Compare checked out objects with last checked in version

When you open the "Pending Objects" form, you see something like this:



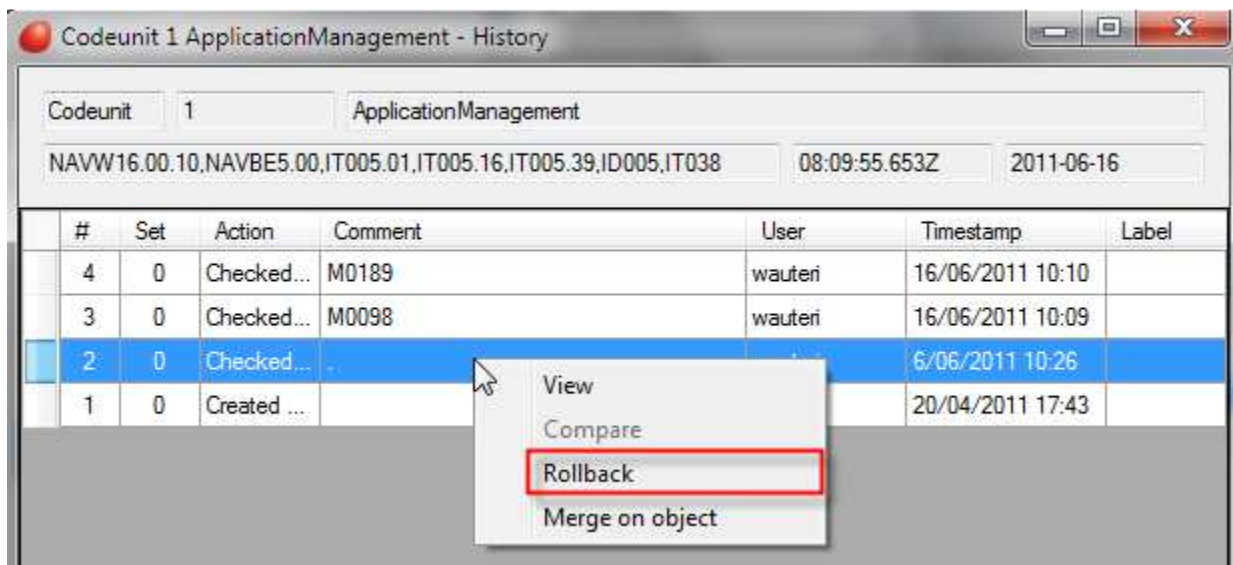
With the buttons “Mark”, “UnMark” and “Marked Only”, you can make a selection of what you want to check in and what not: it’s only going to check in the marked objects.

Furthermore, you’re able to use the comparetool with the buttons “Compare Selection” and “Compare All”. When pressing these buttons, ReVision is going to prepare a compare-session for you with the last checked in version of the object. This way, you can see what you have been changing since the last check in.

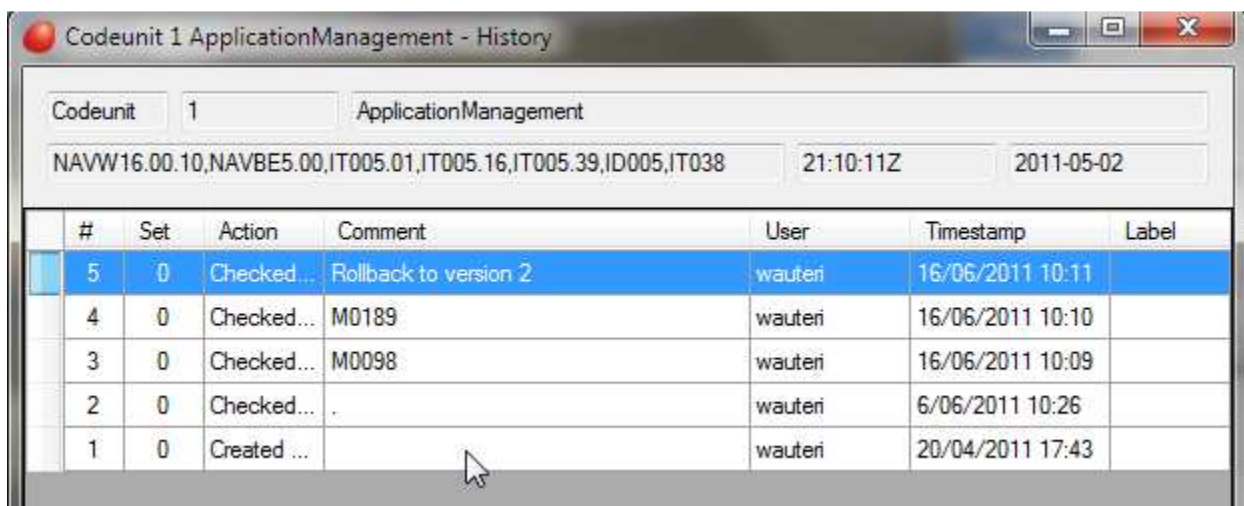
Rolling back Objects

Rolling back an object can be done by using the History table (see “Checking the history of an object”).

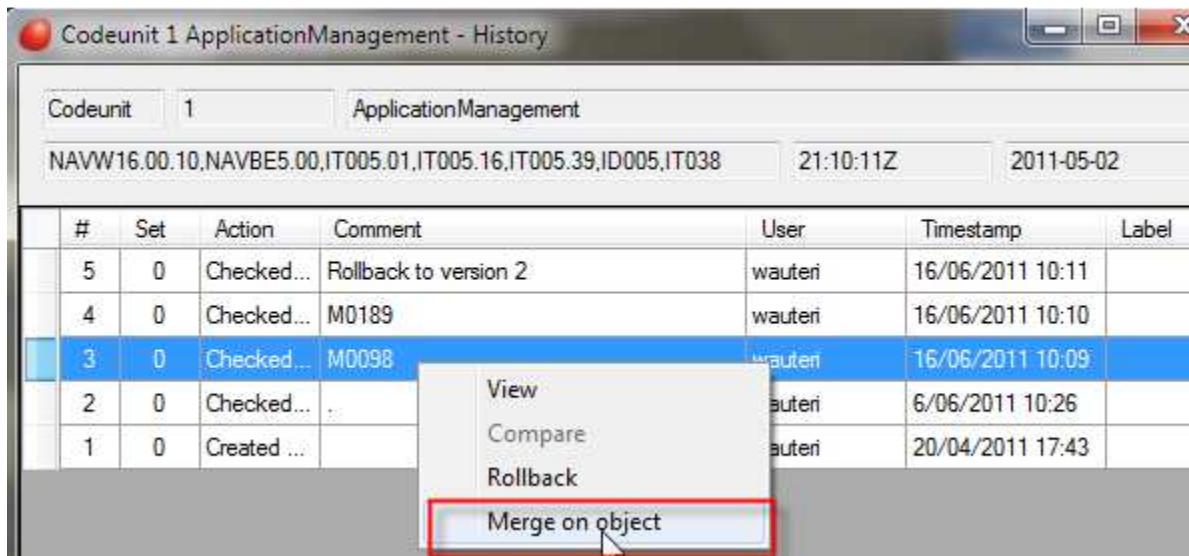
When on the history, you can right-click on a version:



When you click, it will create a new version for you:



It's also possible to merge a certain version with the current version by using the "Merge On Object" option:



It will open a compare window for you. All changes you do in the right window (which is the current version of the object) will be automatically imported (when saved) and compiled (if Live Compile is checked in the preferences).

You can do this “Merge on object” also with two old versions. You’ll get a three way compare.

Note: this works best with Araxis merge. When using Beyond Compare, you’ll have to set it up that it opens a three way compare by default.

Hotkeys

To be able to control your sources more effectively, you have a number of shortcut keys available. The shortcuts only work when a Dynamics NAV client is active, and connected to a database that is added in the repository.

Shortcut	Function
CTRL+ALT+L	Shows the Log
CTRL+ALT+O	Checks out the selected objects in the Object Designer
CTRL+ALT+P	Shows the pending objects (checked out objects)
CTRL+ALT+H	Shows the selected object’s history
CTRL+ALT+I	Checks in the selected objects


Other Development Tools

Open Client

Open Classic Client from ReVision

This is a very handy tool, for which you don't have to do anything .. just use ReVision. Because ReVision has got the database information on Repository-level, and your client information on machine-level. This means it can combine the two and gives you the possibility to open the Client from ReVision.

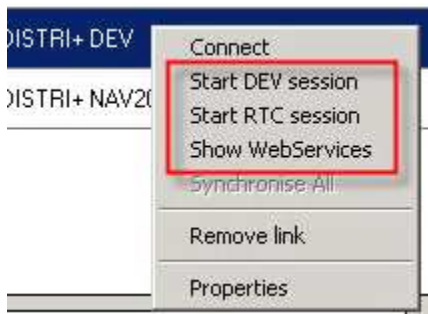
Just select the repository from which you want to start the client.

Dynex		\\10.1.51.14\wauter...	<input checked="" type="checkbox"/>	5.0.26084
 iFacto Default DB	W7-008325	NAV2009R2_iFactoD...	<input checked="" type="checkbox"/>	6.0.6.32198

The last columns is going to show you the NAV Version that used for this repository. The column "NavClient" specifies whether it's found a client on your desktop or not.

It is NOT going to search for clients, so if not found, you only have to open it once, connect to the database you want, and from then on, ReVision will be automatically set up to open it from ReVision from then on.. .

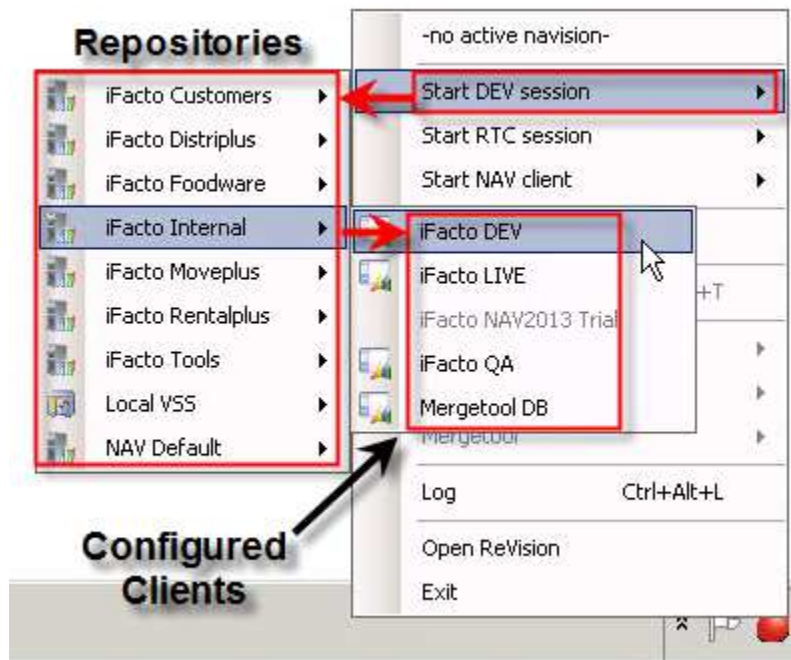
You can do this by double clicking, or right mouse click:



Note: When client is already open, ReVision will focus on the client

When everything is configured. It is possible to open the client by double clicking, or even by right clicking the system tray and click "Open client":

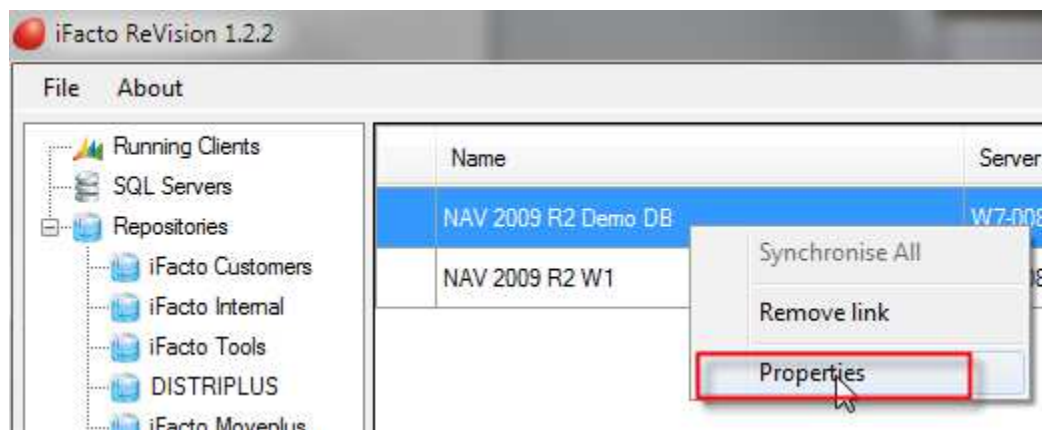
This menu will look the same for all users that configured the repositories, and that way, automatically maintained for all users.



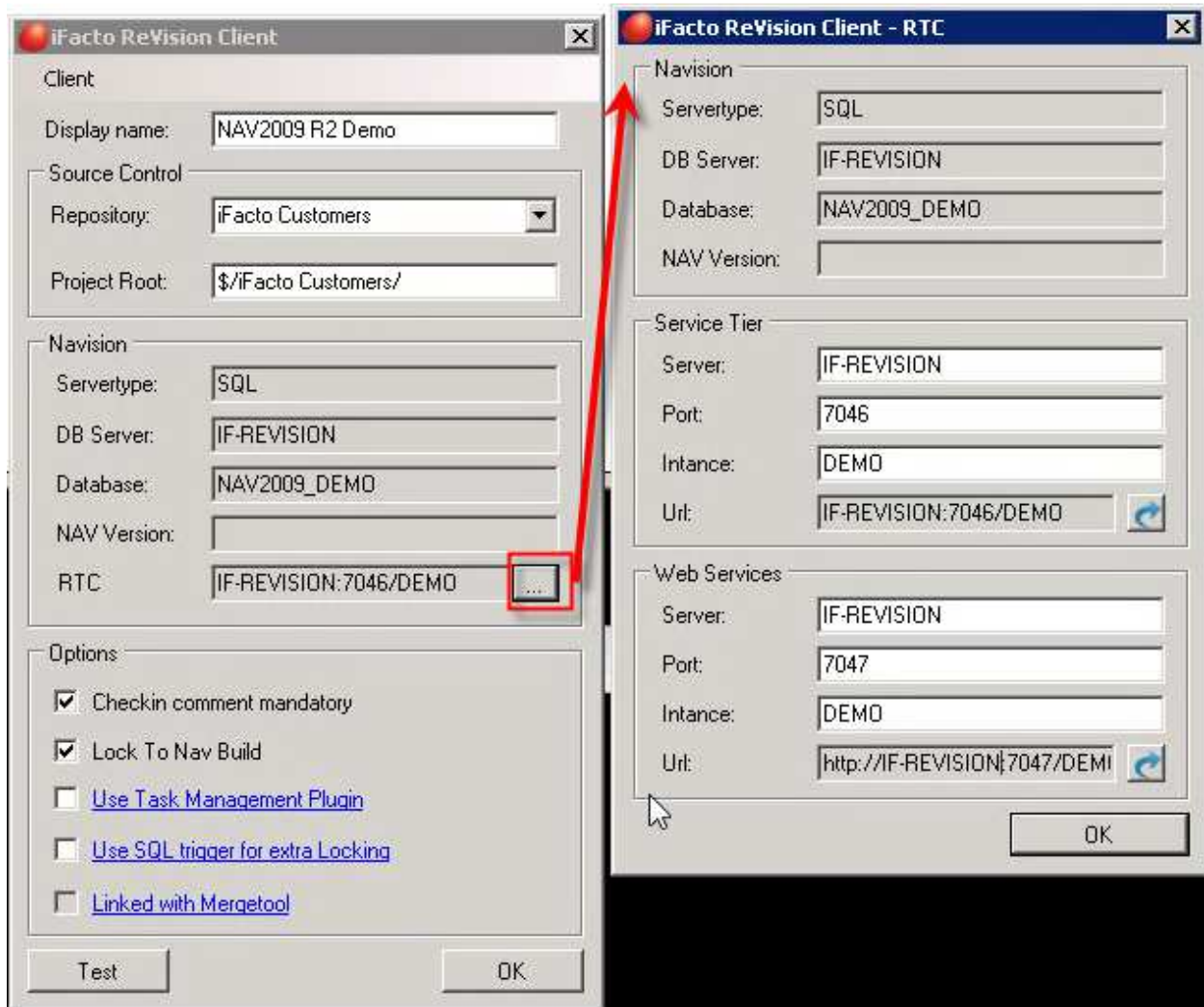
Open RTC Client from ReVision

iFacto ReVision also foresees an option to centrally maintain the RTC connections. Issues with different builds are automatically managed this way, which is a big advantage!

You set up a connection on the properties of a client:

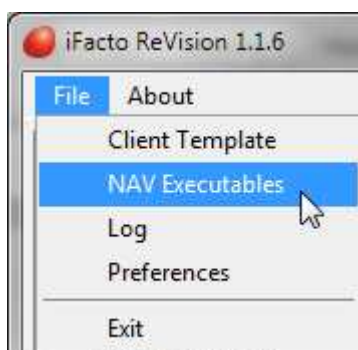


The assistbutton next to "RTC" will guide you to set this up:



Scan for all NAV Executables

When you click File / NAV Executables, you get all configured executables that ReVision is aware of:



iFacto ReVision - Nav Executables		
Scan folder	Open Client	Open Containing Folder
Client	Version	Path
Native	6.0.32012	N:\NAV2009_R2\32012\Classic\fin.exe
SQL	6.0.32012	N:\NAV2009_R2\32012\Classic\fin.sql.exe
RTC	6.0.32012.0	N:\NAV2009_R2\32012\RoleTailored Client\Microsoft.Dynamics.Nav.Client.exe
Native	6.0.33413	N:\NAV2009_R2\33413\Classic\fin.exe
SQL	6.0.33413	N:\NAV2009_R2\33413\Classic\fin.sql.exe
RTC	6.0.33413.0	N:\NAV2009_R2\33413\RoleTailored Client\Microsoft.Dynamics.Nav.Client.exe
Native	6.0.29626	N:\NAV2009_SP1\29626\Classic\fin.exe
SQL	6.0.29626	N:\NAV2009_SP1\29626\Classic\fin.sql.exe
RTC	6.0.29626.0	N:\NAV2009_SP1\29626\RoleTailored Client\Microsoft.Dynamics.Nav.Client.exe
Native	4.0.3.25143	N:\NAV400_SP3\25143\Classic\fin.exe
SQL	4.0.3.25143	N:\NAV400_SP3\25143\Classic\fin.sql.exe
Native	4.0.3.30033	N:\NAV400_SP3\30033\Classic\fin.exe
SQL	4.0.3.30033	N:\NAV400_SP3\30033\Classic\fin.sql.exe
Native	5.0.24199	N:\NAV500\24199\Classic\fin.exe
SQL	5.0.24199	N:\NAV500\24199\Classic\fin.sql.exe
Native	5.0.25653	N:\NAV500\25653\Classic\fin.exe
SQL	5.0.25653	N:\NAV500\25653\Classic\fin.sql.exe

In this window, you get the opportunity to

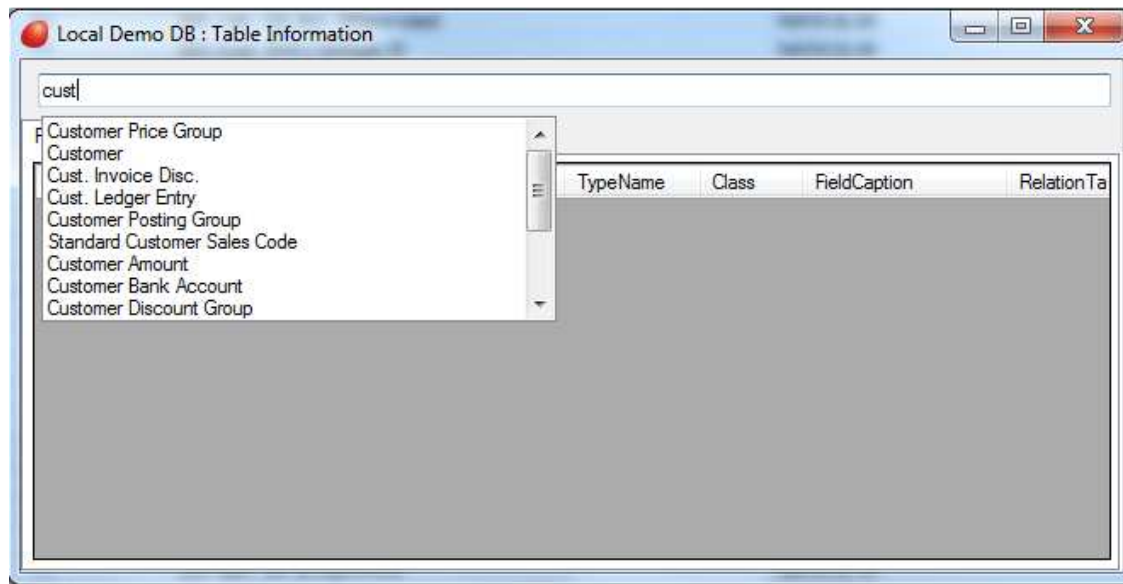
- Scan for any Dynamics NAV executable on your system
- Sort, search and start Dynamics NAV executable on your system.

This is a great way to make ReVision aware of all your client versions and builds in one go.

Table Information

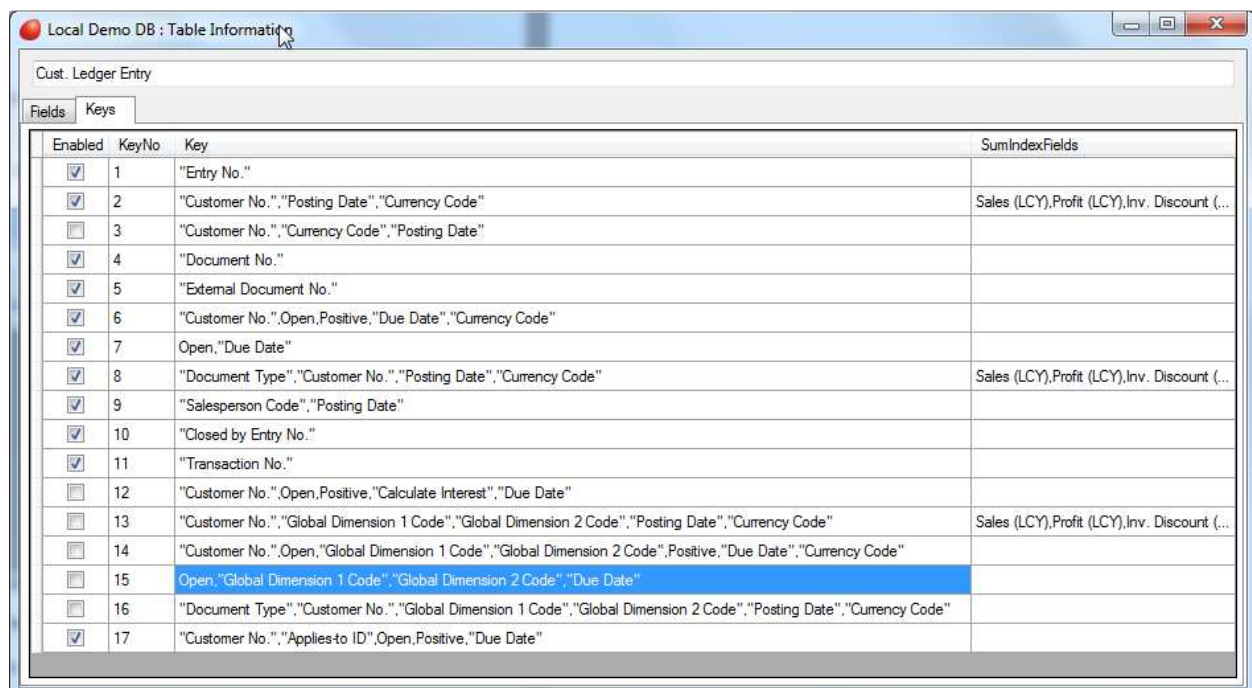
This is a small tool to provide you table information, like field name, keys, ... You can access it from the Tray Icon / Table Information (or CTRL+ALT+T).

You can start typing and it will auto-complete your search:



After selecting a table, it will show you the fields and keys.

Notice that the keys can be directly copied to your code, as the string already contains the necessary "double quotes" (which are necessary in C/AL).



This tool is only here to give you quick information.

External Editor

When you selected an object in the Object Designer, click the Tray Icon / Open Object in Editor. It will open your preferred text editor and opens the object in text-format.

You can edit this object and save. It will import and compile the object.

Note: There are two possible failures:

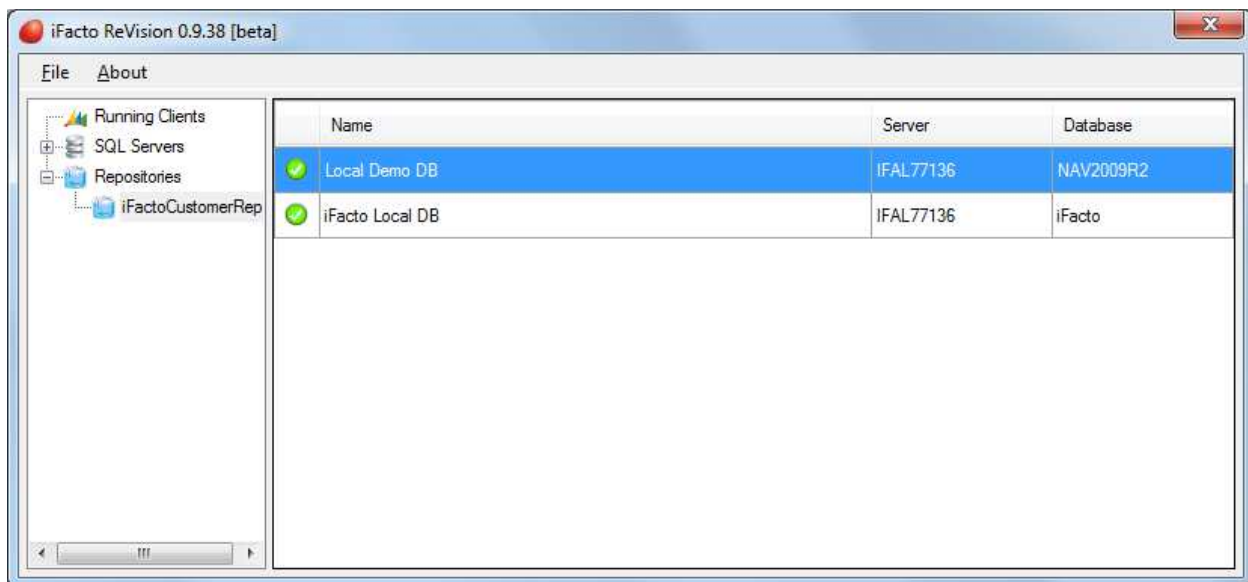
- You're not able to import the file
 - o You made a mistake in the text
 - o The object is not imported and no change to the object is made in NAV
- You're not able to compile
 - o You made a syntax error
 - o The object IS imported in NAV, but uncompiled.

Both situation can have both advantages and disadvantages. Just be aware of it.

Compare & Compile

When you connect to two different databases, you can compare the object of those databases . The only condition is that both are added to a repository.

Consider this situation: I have opened Dynamics NAV two times (different NAV versions):



Suppose I want to compare a number of objects (in this example, I just filtered on "Job*Q*" and selected all objects):

Object Designer

Type	ID	Name	Modified	Version List
Table	278	Job Journal Quantity		NAVW 16.00
Form	470	Job Queue Setup		NAVW 16.00
Report	471	Job Queue Process		NAVW 16.00
Dataport	472	Job Queue Entry		NAVW 16.00
XMLport	473	Job Queue Response		NAVW 16.00
Codeunit	474	Job Queue Log Entry		NAVW 16.00
MenuSuite	670	Job Queue Setup		NAVW 16.00
Page	671	Job Queue Processes		NAVW 15.00
	672	Job Queue Entries		NAVW 15.00
	673	Job Queue Entry Card		NAVW 15.00
	674	Job Queue Log Entries		NAVW 15.00
	675	Job Queue Process Priority		NAVW 15.00
	448	Job Queue Dispatcher		NAVW 16.00
	449	Job Queue Start Codeunit		NAVW 15.00.01
	670	Job Queue Setup		NAVW 16.00
	671	Job Queue Processes		NAVW 15.00
	672	Job Queue Entries		NAVW 15.00
	673	Job Queue Entry Card		NAVW 15.00

Right-click the Tray Icon / Compare / YourOtherOpenedNAVClient . It will export all objects of both environments and open your compare tool. As you can see, in my example, only one object was changed:

Araxis Merge - [C:\Users\...304\iFacto Local DB\ and C:\Users\...21304\Local Demo DB\]

Folder	Changes	Folder
Codeunit		Codeunit
Codeunit 448 Job Queue Dispatcher.txt	1	Codeunit 448 Job Queue Dispatcher.txt
Codeunit 449 Job Queue Start Codeunit.txt	0	Codeunit 449 Job Queue Start Codeunit.txt
Form		Form
Form 670 Job Queue Setup.txt	0	Form 670 Job Queue Setup.txt
Form 671 Job Queue Processes.txt	0	Form 671 Job Queue Processes.txt
Form 672 Job Queue Entries.txt	0	Form 672 Job Queue Entries.txt
Form 673 Job Queue Entry Card.txt	0	Form 673 Job Queue Entry Card.txt
Form 674 Job Queue Log Entries.txt	0	Form 674 Job Queue Log Entries.txt
Form 675 Job Queue Process Priority.txt	0	Form 675 Job Queue Process Priority.txt
Page		Page
Page 670 Job Queue Setup.txt	0	Page 670 Job Queue Setup.txt
Page 671 Job Queue Processes.txt	0	Page 671 Job Queue Processes.txt
Page 672 Job Queue Entries.txt	0	Page 672 Job Queue Entries.txt
Page 673 Job Queue Entry Card.txt	0	Page 673 Job Queue Entry Card.txt
Page 674 Job Queue Log Entries.txt	0	Page 674 Job Queue Log Entries.txt
Page 675 Job Queue Process Priority.txt	0	Page 675 Job Queue Process Priority.txt
Table		Table
Table 278 Job Journal Quantity.txt	0	Table 278 Job Journal Quantity.txt
Table 470 Job Queue Setup.txt	0	Table 470 Job Queue Setup.txt
Table 471 Job Queue Process.txt	0	Table 471 Job Queue Process.txt
Table 472 Job Queue Entry.txt	0	Table 472 Job Queue Entry.txt
Table 473 Job Queue Response.txt	0	Table 473 Job Queue Response.txt
Table 474 Job Queue Log Entry.txt	0	Table 474 Job Queue Log Entry.txt

Press F1 for help

0 removals · 0 insertions · 1 change

Furthermore, you can open this object, and see the changes.

When comparing, it's possible to change the right-side object (the object where you wanted to compare with .. of the selected client in your tray icon). When saving this text file, iFacto ReVision is picking up the change, and will try to import it into this client. The object will be compiled (when possible).

Note

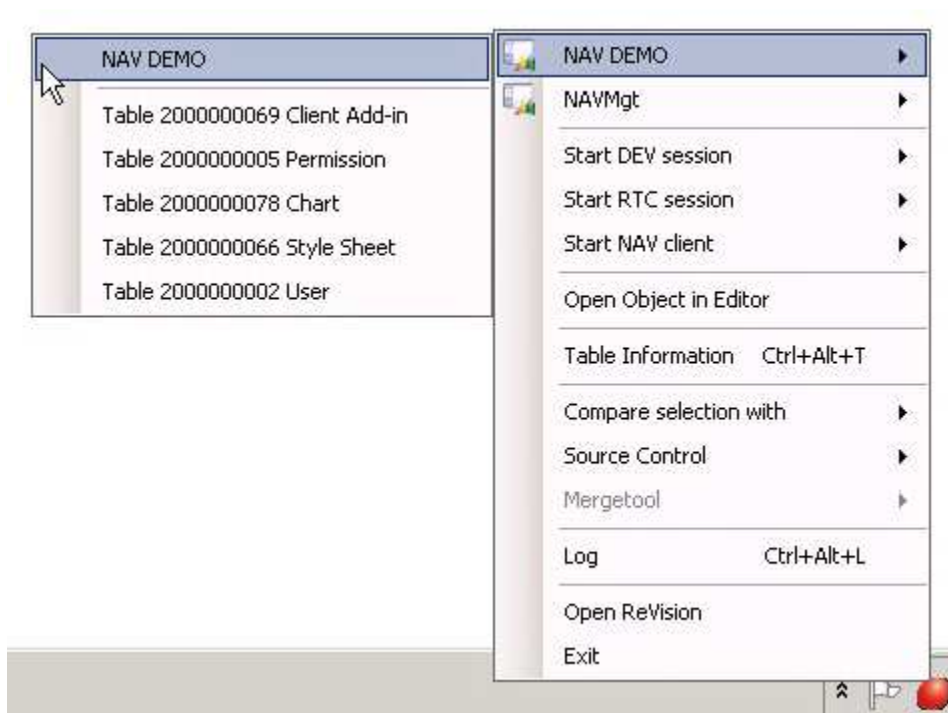
This tool is extremely useful when:

- You compare a DEV and TEST or TEST and LIVE environment.
- You only want to merge parts of developments.

Focus windows from Tray Icon

During development, it can (and will) happen that multiple design windows are open in your client. Searching for the necessary window is not always that easy.

iFacto ReVision keeps track of these windows, and give you the possibility to focus the right window in the right NAV client.



In the above situation, I have opened two clients "NAV DEMO" and "NAVMgt". In the first one, I have opened 5 design windows. Selecting one of these windows will focus the NAV client, and focus the window inside the client.

Link with Mergetool

iFacto Revision has a tight integration with Mergetool (www.mergetool.com). It can create versions for you, open a version, open contrasts, ..

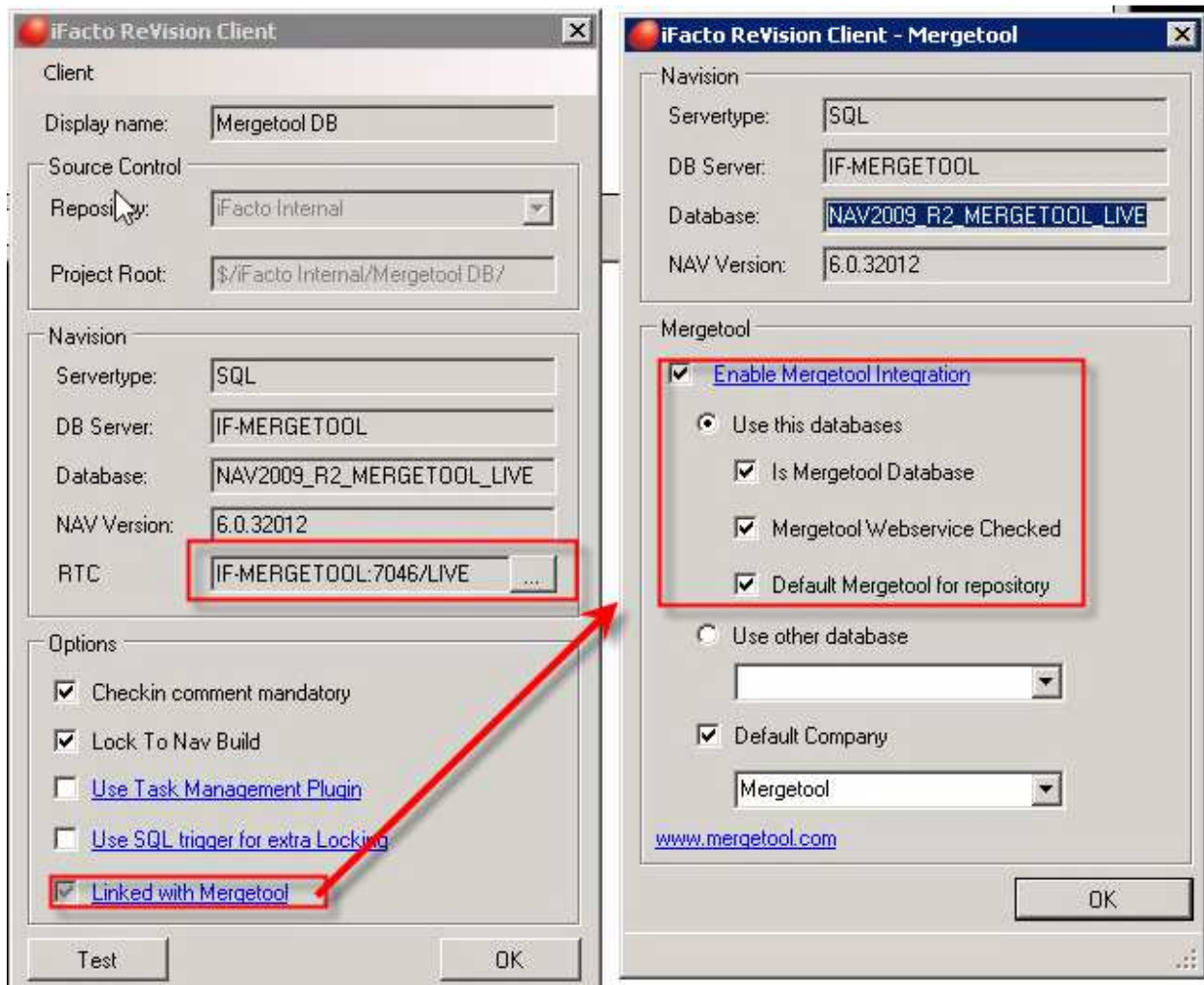
Setting up a central Mergetool Database

First of all, you must set up a new Mergetool Database, in RTC, where you publish following webservice:

Object T...	Service Name	Object ID	Published
Codeunit	MGT_Service	14120009	✓

Open the Database and add it to a repository. Next, you can set up its repository properties, and set up the Mergetool properties:





It's important to set up the RTC settings for the interaction between iFacto ReVision and Mergetool. Furthermore, you should set up this database as being a mergetool database, by clicking "Linked with Mergetool and clicking "enable Mergetool Integration":

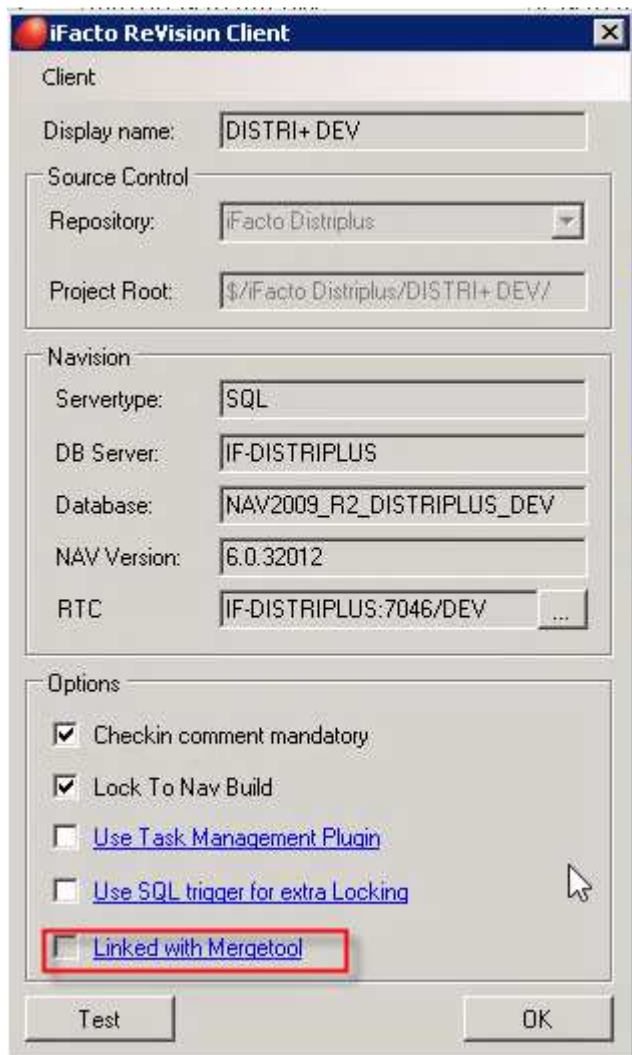
This last link is going to check the database as it meets the standards for an integration:

- Does it contain the Mergetool objects
- Is the webservice available

From now on, you're good to go to link other databases to this central mergetool database

Link a DEV database to a central Mergetool database

You obviously do this in the properties of the database you want to link, by clicking again the "link with mergetool" link:

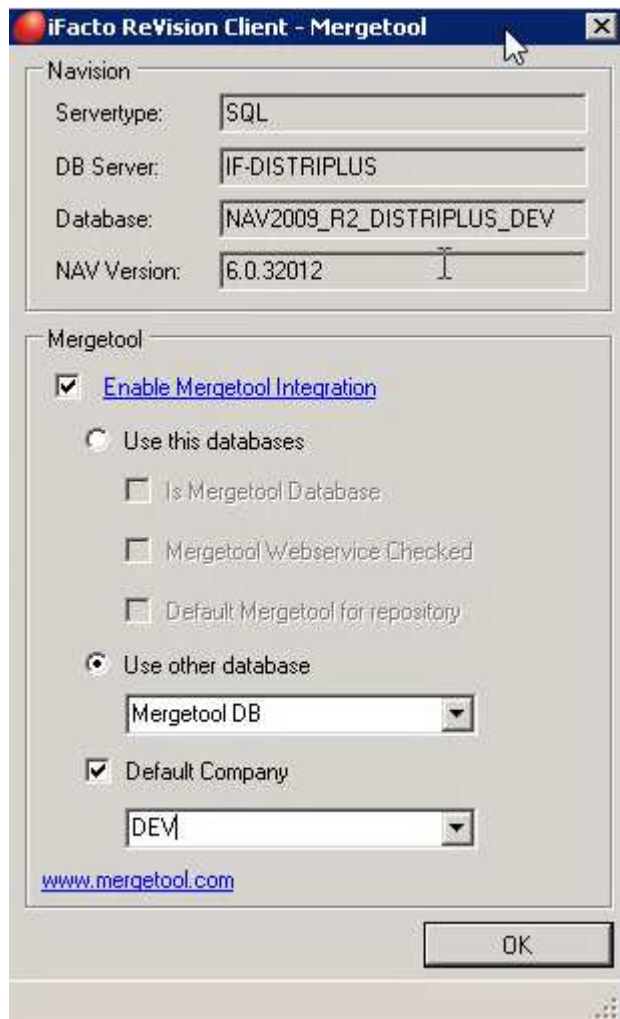


The image shows the 'iFacto ReVision Client' dialog box. It has a title bar with the iFacto logo and the text 'iFacto ReVision Client'. The dialog is divided into several sections: 'Client', 'Source Control', 'Navision', and 'Options'. In the 'Client' section, 'Display name' is 'DISTRI+ DEV'. In the 'Source Control' section, 'Repository' is 'iFacto Distriplus' and 'Project Root' is '\$/iFacto Distriplus/DISTRI+ DEV/'. In the 'Navision' section, 'Srvtype' is 'SQL', 'DB Server' is 'IF-DISTRIPLUS', 'Database' is 'NAV2009_R2_DISTRIPLUS_DEV', 'NAV Version' is '6.0.32012', and 'RTC' is 'IF-DISTRIPLUS:7046/DEV'. In the 'Options' section, there are five checkboxes: 'Checkin comment mandatory' (checked), 'Lock To Nav Build' (checked), 'Use Task Management Plugin' (unchecked), 'Use SQL trigger for extra Locking' (unchecked), and 'Linked with Mergetool' (unchecked). The 'Linked with Mergetool' checkbox is highlighted with a red rectangle. At the bottom are 'Test' and 'OK' buttons.

Section	Field	Value
Client	Display name:	DISTRI+ DEV
	Repository:	iFacto Distriplus
Source Control	Project Root:	\$/iFacto Distriplus/DISTRI+ DEV/
	Srvtype:	SQL
Navision	DB Server:	IF-DISTRIPLUS
	Database:	NAV2009_R2_DISTRIPLUS_DEV
	NAV Version:	6.0.32012
	RTC:	IF-DISTRIPLUS:7046/DEV
	Options	Checkin comment mandatory
	Lock To Nav Build	<input checked="" type="checkbox"/>
	Use Task Management Plugin	<input type="checkbox"/>
	Use SQL trigger for extra Locking	<input type="checkbox"/>
	Linked with Mergetool	<input type="checkbox"/>

When you click the link “Enable Mergetool Integration” now, ReVision will notice it’s not a Mergetool database, and will set it up as a link right away. All you have to do, is link it with one of your set up central Mergetool databases.

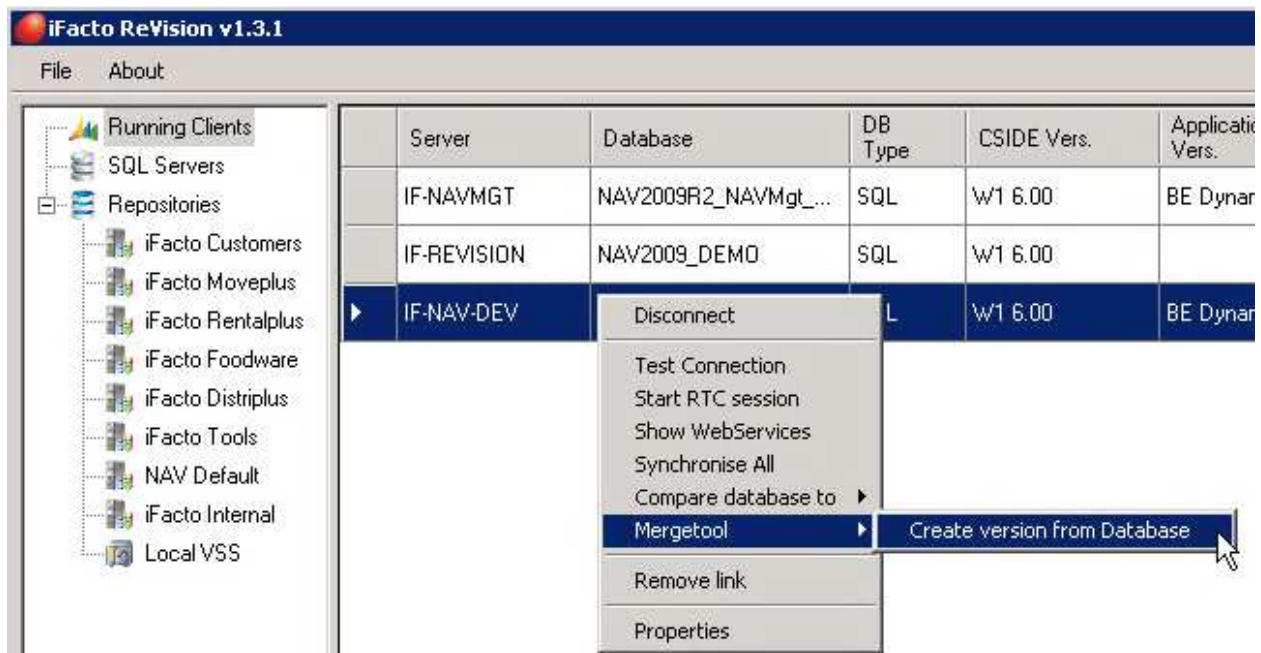
for your convenience, you can set up a company in Mergetool for this database as well:



How to use the link with Mergetool

Mergetool is all about creating, exploring, analyzing, maintaining, ... versions of objects. With this integration, you can easily upload objects to versions in Mergetool.

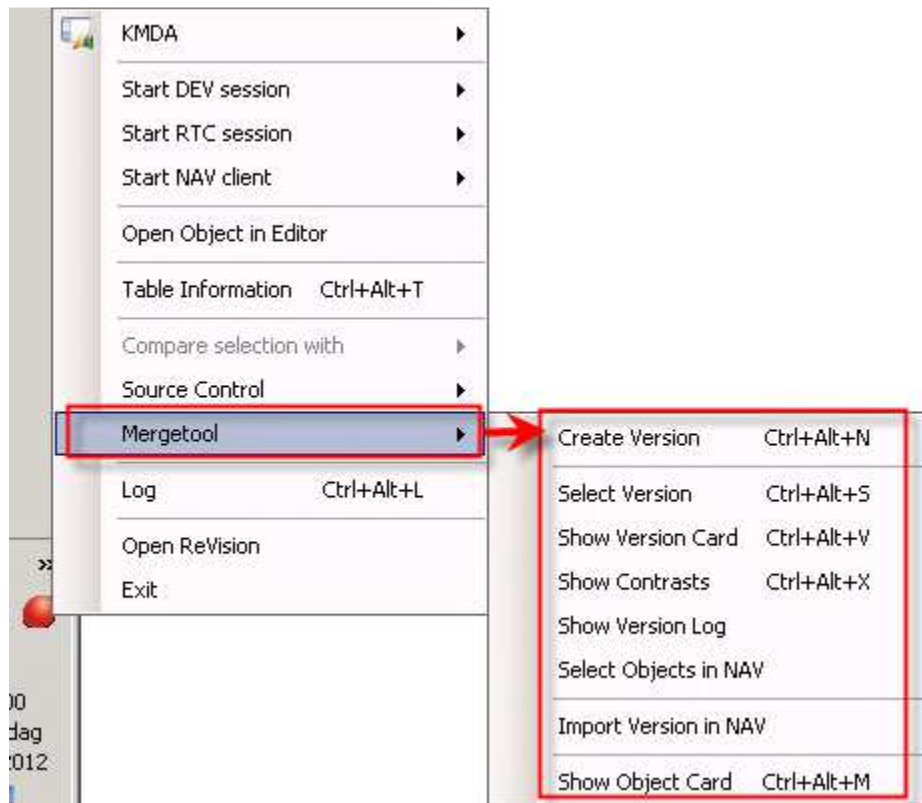
The most-used option is most probably the “create version from Database”:



By doing so, ReVision is going to upload all objects to Mergetool. If you check the “Run SourceCode analysis”, Mergetool will analyze all relations between the objects for you.

After the upload, ReVision will open the Version Card for you.

Next to this upload, there are some smaller functions for you in the system tray menu:



Function	Description
Create Version	Create a version from a limited set of objects (the ones you selected in the object designer)
Select Version	Make a version in the Mergetool database the current version to work with from ReVision. You use this for example before importing a version from mergetool into NAV.
Show Version Card	Open the currently selected version card.
Show Contrasts	Open the contrasts from the currently selected version.
Show Version Log	Open the version log from the currently selected version.
Select Objects in NAV	Select all objects in NAV which were uploaded in the currently selected version.
Import Version in NAV	Import all objects in NAV which were uploaded in the currently selected version.
Show Object Card	Show the object card from the selected object in the Object Designer, from the version which is selected in Mergetool.