

# Introduction to NavRepository

**NavRepository** is a powerful simple-to-use repository tool for *MS Dynamics NAV*. It eases development and maintenance of *Navision* objects by providing local repository for *NAV* objects, and thus allowing integration of *NAV* with external version control systems, such as *SVN* or *CVS*. **NavRepository** makes easier the daily work of *NAV* developers and users.

Here are some major features of **NavRepository**:

- Provides local repository for *Navision* objects
- Allows *NAV* integration with external version control systems, such as *SVN* or *CVS*
- Provides various options for object synchronization
- Reduces the the user interaction to only two main actions
- Provides possibility to change the objects' synchronization actions
- Keeps history of the local changes
- Tracks modifications in object definitions
- Provides log of changes and possibility to revert to history versions
- Provides a service to create a web model of the database

## Local Repository for Navision Objects

**NavRepository** creates a local repository for the *Navision* objects (forms, reports, pages, etc.). This way unintentional modifications of the synchronized objects could be easily reverted. On the other hand, after the modifications of the database objects are tested and approved, these changes could be saved into the local repository. The repository objects could be then exported and sent to the other interested users, developers or consultants or committed to a dedicated version control system.

Objects							
*	Type	ID	Name	M	C	Status	
	Form	1	Company Information	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Unchanged	
	Form	14	Salespeople/Purchasers	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Unchanged	
	Form	16	Chart of Accounts	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Unchanged	
	Form	20	General Ledger Entries	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Unchanged	
	Form	21	Customer Card	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Unchanged	
	Form	22	Customer List	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Unchanged	
	Form	25	Customer Ledger Entries	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Unchanged	
	Form	26	Vendor Card	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Unchanged	

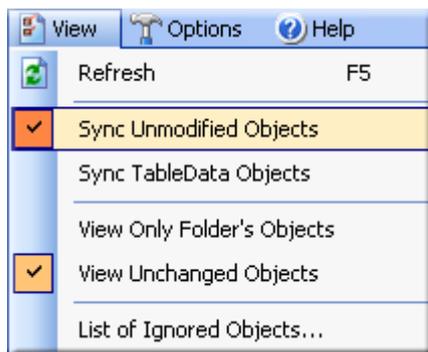
## Integration with External Version Control Systems

The local repository, created by **NavRepository** is designed in a way to be easily accessible by version control systems, such as *SVN* or *CVS*. This way the modifications of the local objects could be instantly propagated to multiple development locations or to multiple end-users.

Name	Revi...	Local State	Copy From	Relative Dir
1.blb	12	Unchanged		FORM
11080.blb	12	Unchanged		FORM
14.blb	12	Unchanged		FORM
16.blb	12	Unchanged		FORM
20.blb	12	Unchanged		FORM
21.blb	12	Unchanged		FORM
22.blb	12	Unchanged		FORM
25.blb	12	Unchanged		FORM
26.blb	12	Unchanged		FORM

### Options for Synchronization of the Objects

**NavRepository** provides various options for synchronization of the *Navision* objects. You can synchronize only the database objects with Modified flag turned on or both modified and unmodified objects. You can choose whether to synchronize the table-data objects or not. You can opt whether to synchronize the textual object definitions (the object's source code) or just the BLOB references.



### Simplicity of Use - Only Two Main Actions

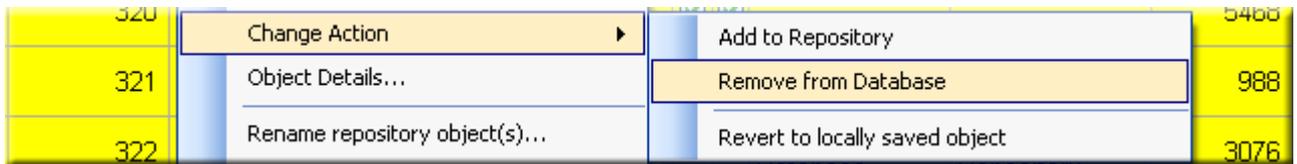
The user interaction in **NavRepository** is the simplest possible. Tracking of objects' modifications is done automatically after the start-up or manually by pressing the *Refresh*-button. The needed synchronization actions for each object are suggested automatically. The practical user interaction is reduced to only two main actions - commit the modifications or revert the changes.

ID	Name	M	C	Status	Action	Size
318	Commit actions	✓	✓	Added to Database	Add to Repository	988
319	Commit actions and export	✓	✓	Added to Database	Add to Repository	1604
	Revert to locally saved objects					

### Changing the Object's Synchronization Action

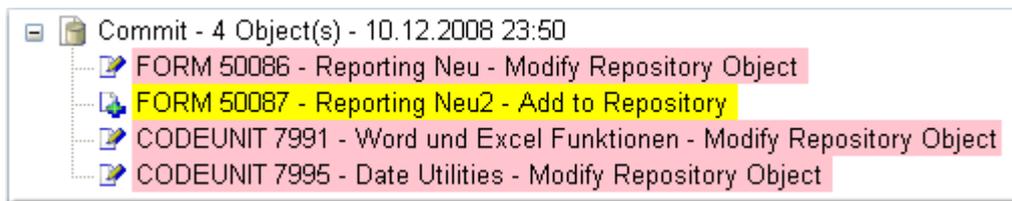
Synchronization actions for each object are suggested automatically. Anyway, the user has the final word. He can commit the modification (i.e. execute the synchronization action as

suggested), revert the modification, or change the synchronization action to another available action for this type of modification. This is especially useful when there is a conflict (i.e. both the database and repository copies of the object are modified). **NavRepository** does its best to suggest the correct synchronization action, but we encourage the users to check always the suggested actions and change them, if they seem incorrect.



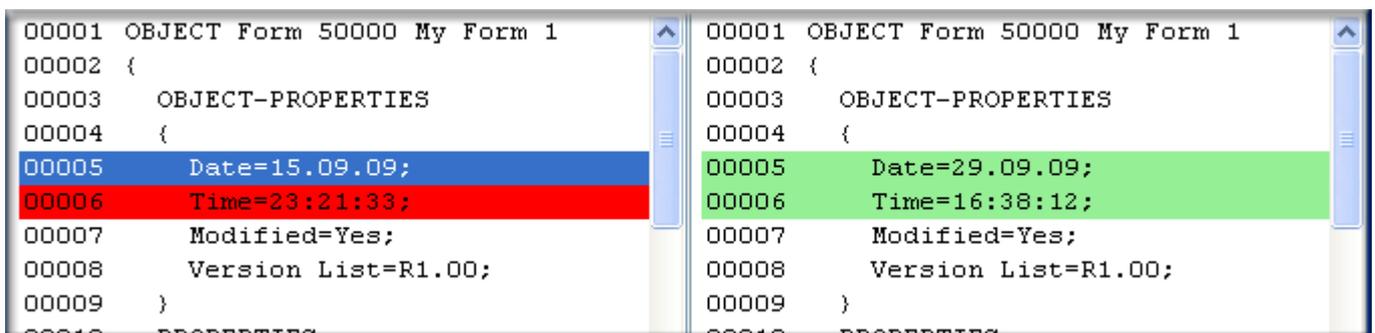
## History of the Local Changes

**NavRepository** keeps history logs of the local changes and shows the latest changes on the main screen. The history of batch actions shows the user actions (commits or reverts), and the batch of objects affected by that action, including the object's status before that and the action executed. It is possible to revert an object or the whole batch of objects to that history version.



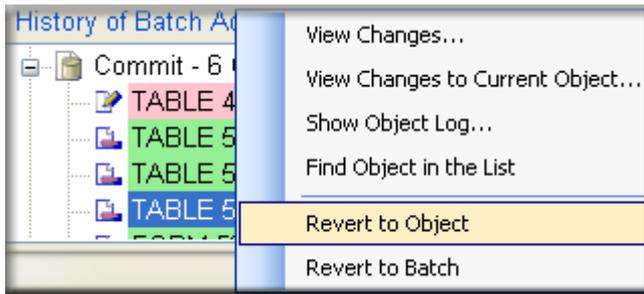
## View Modifications of Object Definitions

**NavRepository v. 2.0** opts to synchronize the object definitions, instead of only the BLOB references. Synchronization of object definitions and preserving the history versions of all synchronized objects makes it possible to provide information about the changes in the object definition, both compared to the previous version and to the most actual (database) version of the object.



## View the log of object's changes

Synchronization of object definitions and preserving the history versions of all synchronized objects makes it possible to provide the full history of object's changes - the object log. The object could be reverted to its history version alone or altogether with the entire batch of objects (all objects synchronized with that action).



## Web Model of the Database

**NavRepository** provides interface to a service, used to create a web model of the database. The model currently includes the tables and codeunits. The code behind the table triggers as well as the functions contained within the tables and codeunits are listed in the model and are syntax colored. This makes the tracking of complex table logic, relations or possible issues easier. The web model of the database could be easily uploaded to a web server and made available to all interested users, developers or consultants.

More information about the web-model service you can find [here](#)

The sample web model of the *NAV Demo Database* you can check out [here](#)

