

CGI GROUP INC.

Code Review Tool for Dynamics AX

User guide

Contents

1. Overview.....	2
2. Installation	2
2.1 Code review pack viewer	2
2.2 Dynamics AX add-on	2
2.3 Setup	3
3. Process.....	6
4. Cookbook	8
4.1 You would like to have somebody review your code as soon as possible	8
4.2 You would like to send a pre-checkin code review	9
4.3 When checking-in last time, SMTP server failed, so no code review pack was sent. You would like to resend the change for review.	10
4.4 You would like to review changes done by one of your colleagues on the same project.....	11
4.5 You are working on your own dev box with AX 2012 at the customer network. You would like to avoid re-entering all email addresses to code review tool address book after each model store import or re-installation of AX	11
4.6 How to conduct a successful code review?	11
4.7 You received a code for review, and it looks like the developer has removed a lot of methods from a standard AX class	11
4.8 Your team works on the same environment in AX 2009. You don't want the code review tool to be part of the XPO you deliver to customers.....	11
4.9 You got feedback in form of a text file, which contains lines numbers and comments	11
4.10 What kind of feedback should be provided to code author?	11

"Peer review – an activity in which people other than the author of a software deliverable examine it for defects and improvement opportunities -- is one of the most powerful software quality tools available."

Karl Wiegers, "The Soft Side of Peer Reviews"

1. Overview

The Code Review Tool is developed by CGI Danmark A/S. The tool is supposed to be used by Dynamics AX application developers and development leads, when implementing customizations and fixing bugs.

The main idea behind the proposed tool is automation of whatever can be automated in the code review process.

The tool consists of 2 parts.

The **first part** is an add-on to standard Dynamics AX. It is imported to AOT as an XPO file. The add-on reads changes from the version control system (VCS) and generates files named "code review packs". Code review packs are then automatically sent to reviewers by e-mail, either manually before a checkin, or automatically after the checkin.

Code review pack files have extension .CRP, but in fact they are ZIP-files. Inside CRP-files, there are 2 folders, named "Before" and "After". Each of the two folders contains one or more XPOs, one per AOT object.

The AX add-on is recommended to be installed on all development environments on the project.

The **second part** of the tool is a stand-alone Windows application. It is supposed to be installed on both the user laptop (where emails are received) and every development environment where the AX add-on (the first part) is installed.

This application will recognize code review packs and open them in a standard graphical file-comparison program.

2. Installation

2.1 Code review pack viewer

Run the *SetupCRPViewer.msi* file on both email laptop and Dynamics AX development machines. The installed application will appear in *Control Panel\Programs\Programs and Features* as *Code Review Pack Viewer*.

Important: on the development machine, the *Code Review Pack Viewer* must be installed **before** the AX add-on – there is a .NET assembly (DotNetZip Library) that must be registered in GAC before the AX add-on is imported to the AOT.

2.2 Dynamics AX add-on

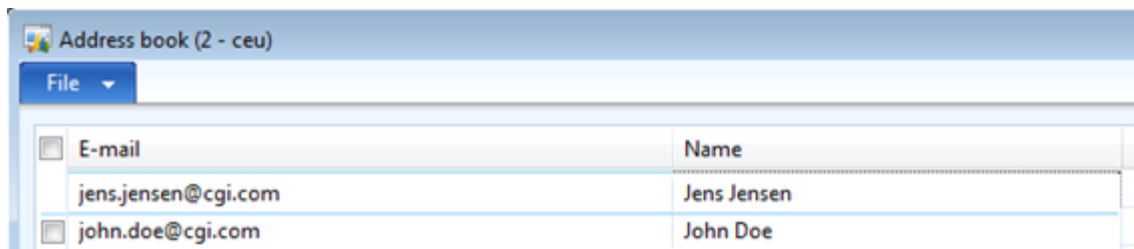
1. Turn off version control in AX.
2. In the import dialog, select the XPO, corresponding to your version of Dynamics AX, and click on "Show details" checkbox. Take a look at the objects that will be imported. There are several standard objects, mostly version control forms. If any of them are already customized at your installation, carefully merge the changes, otherwise proceed to step 3.
3. Click OK to finish the import.
4. Restart the AX client.
5. Turn on the version control.

2.3 Setup

In the AX development workspace menu, click *Tools > Code review > Address book*.

In the Address book form, add names and emails of all parties that might be involved in the code review process. You will need to add the following emails:

- yourself,
- technical responsible and/ or development lead,
- one or more developers working on the same project.

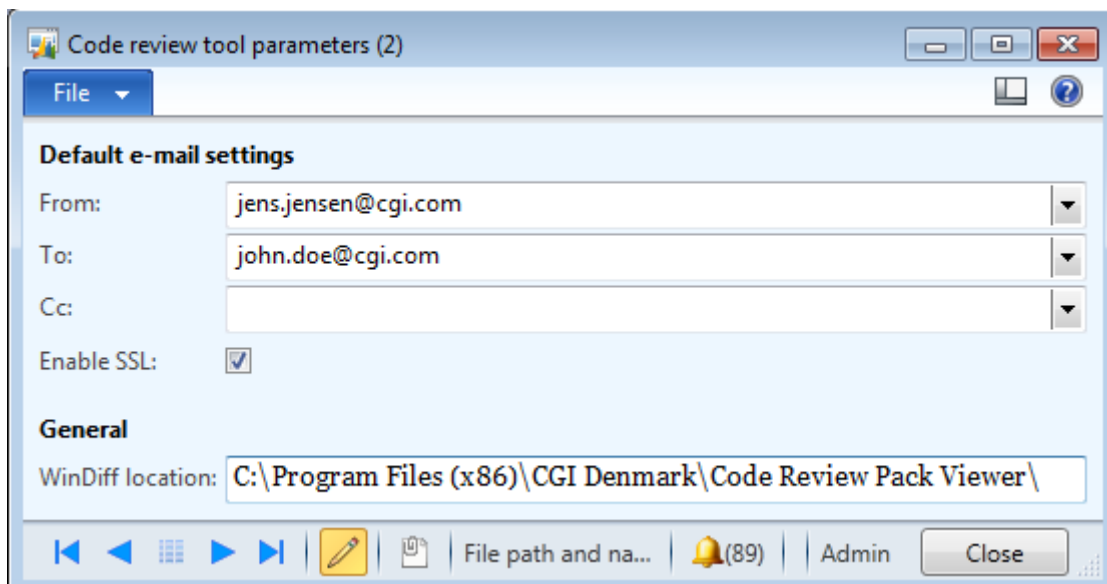


Important: jens.jensen@cgi.com and john.doe@cgi.com are fake addresses used here for demonstration purposes. Use your actual corporate e-mail addresses in the real environment.

Now, in the AX development workspace menu, click *Tools > Code review > Code review tool parameters*.

Here, you may select default email parameters and path to the *WinDiff* tool.

In the following example, Jens Jensen is the developer sending code for review (you should select your own email in the From field). John Doe is his fellow developer, who will do the actual review.



WinDiff.exe is a standard Microsoft comparison tool. It is rather old, but easy to use and should be enough for our tasks. (If you are not familiar with it, try comparing 2 text files, turning on and off different options in the menu, and see the effect.)

You should make sure that *WinDiff.exe* location parameter is set up, so the code review tool may run *WinDiff*, when needed. You can find the *WinDiff.exe* file in *C:\Program Files (x86)\CGI Denmark\Code Review Pack Viewer*, after the Code Review Pack Viewer is installed.

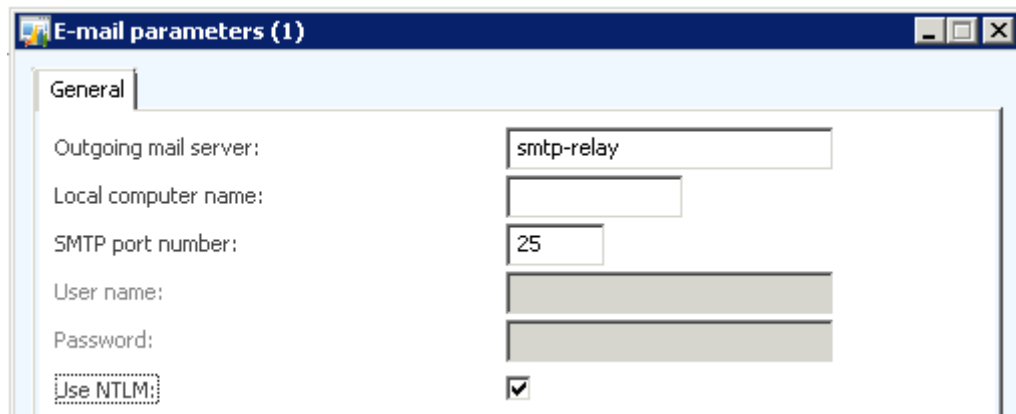
The last thing to set up is email parameters in the standard *System Administration > Setup > System > E-mail parameters* form.

There are 2 options here:

- **Use the corporate SMTP relay server.** This is the preferable option. If you don't know the server name and port number, you can ask you AX helpdesk or IT helpdesk for assistance.

If you are external consultant, it will most likely be credentials given to you by the customer IT department. The credentials may either be explicit or implicit (as on the following screenshot). So, in the *From* field of incoming messages, reviewers will see something like *Jens Jensen <jensj@bigtrucks.com>* (where "Big Trucks" is the customer company name), rather than *Jens Jensen <jens.jensen@cgi.com>* (if you are employed by CGI Group Inc. in our example). However, email headers will contain CGI address in Reply-To property, so clicking *Reply* button will send review feedback to the **actual** CGI email address (*jens.jensen@cgi.com* in our example).

E.g.:

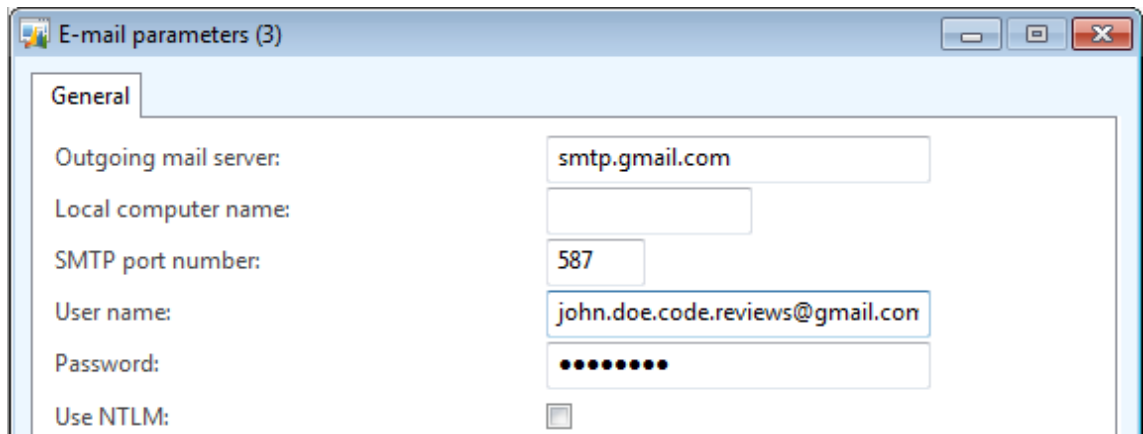


The screenshot shows a Windows-style dialog box titled "E-mail parameters (1)". It has a "General" tab selected. The fields are as follows:

Field	Value
Outgoing mail server:	smtp-relay
Local computer name:	
SMTP port number:	25
User name:	
Password:	
Use NTLM:	<input checked="" type="checkbox"/>

- **There is no SMTP relay server at the customers.** In that case, you will need some external server. For example, you may create an account on *Gmail.com*, and use it for sending emails. However, it may happen that an external mail server can't be used for some reason (there may be issues with establishing connection with external SMTP servers from a corporate network because of firewalls). In that case, you may try to solve the issue with your IT department.

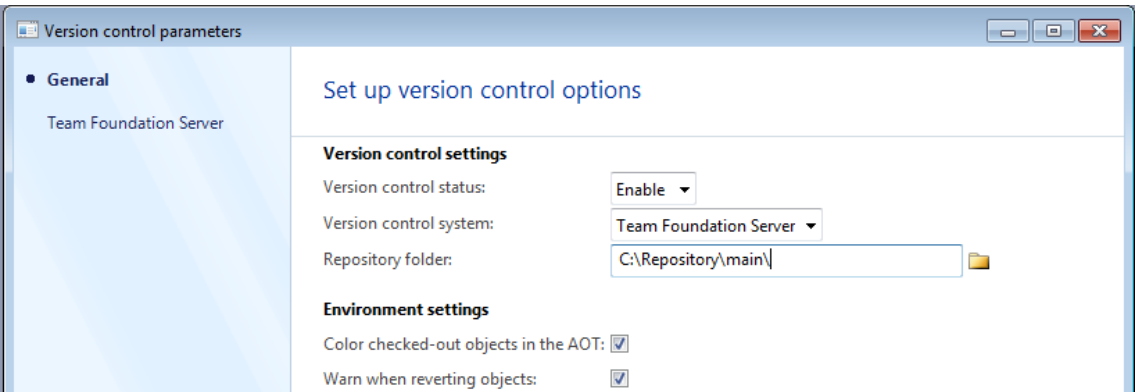
E. g.:



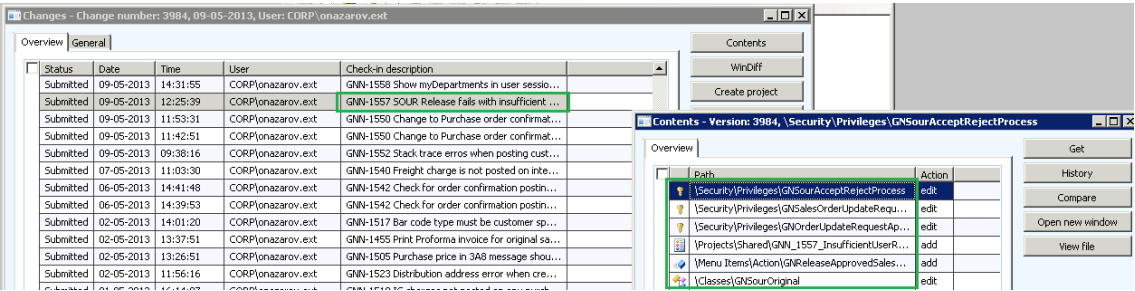
The screenshot shows a Windows-style dialog box titled "E-mail parameters (3)". It has a "General" tab selected. The fields are as follows:

Field	Value
Outgoing mail server:	smtp.gmail.com
Local computer name:	
SMTP port number:	587
User name:	john.doe.code.reviews@gmail.com
Password:	••••••••
Use NTLM:	<input type="checkbox"/>

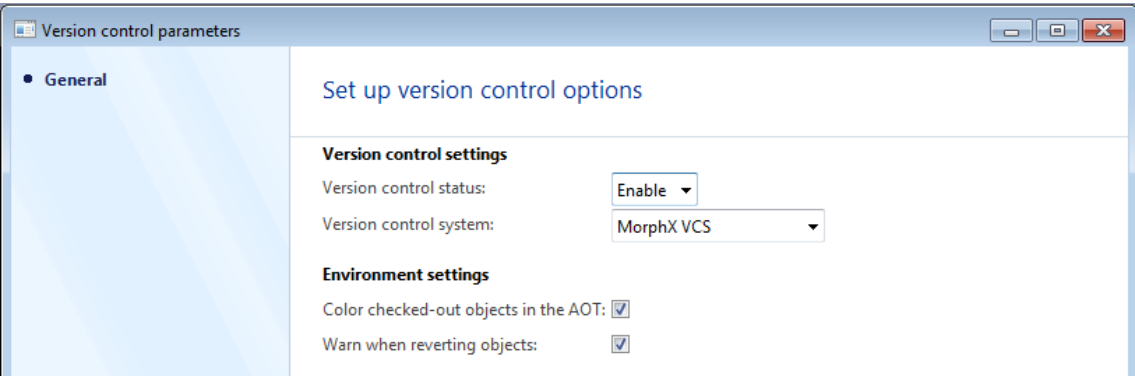
The last step is to make sure that version control is enabled on you development environment. Ideally, it should be TFS version control system (VCS):



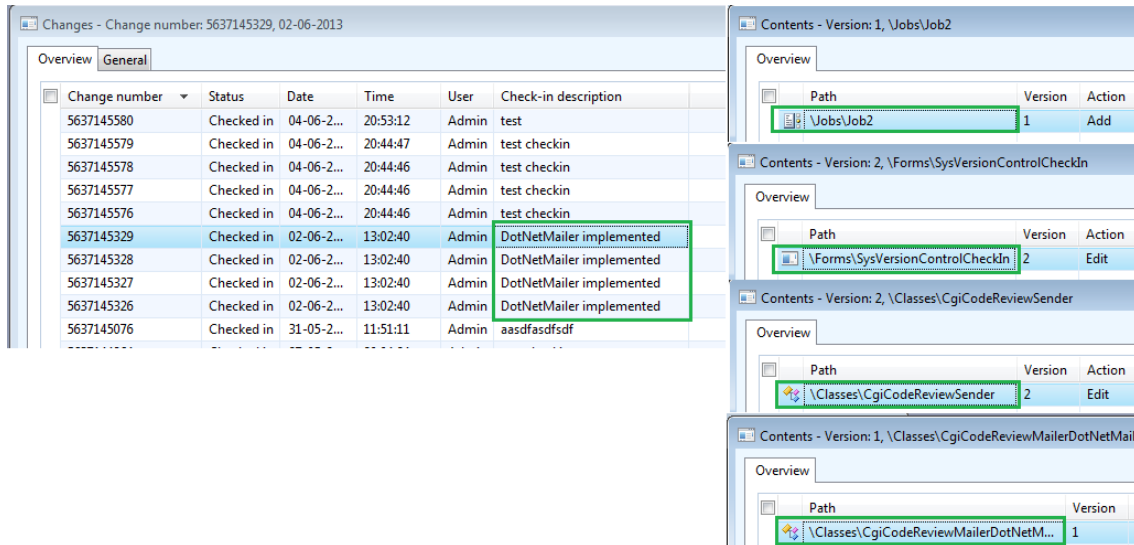
In that case, each change list will be stored as a single entry in the history of changes, and the list of modified objects will be available via Contents button:



If TFS is not available, you should enable MorphX VCS:



MorphX VCS creates one or more entries in the changes history per change list, and Contents form will always show one record:



The code review tool supports both TFS and MorphX version control systems, despite the difference in storing history of changes.

After the code review tool is installed, you will need to verify it is working. Use your own corporate email address in the From and To fields. Check out a couple of AOT objects and make some changes, e.g. add comments. Send a pre-checkin code review to yourself. Wait for the email in your inbox and double-click on the attachment to see the change in WinDiff. Press "Reply" button and send a sample feedback to yourself. Eventually, undo check-out of the objects.

After the setup is verified, set proper email addresses in the code review tool parameters. Now you are ready to send code for review and provide feedback. Remember to tell the reviewers, that they need the Code Review Pack Viewer installed in order to see your changes.

3. Process

Changes may be sent either before or after checkin, or both.

The simplest scenario could look like that:

1. Developer Jens Jensen checks out one or more objects in the AOT and makes his changes.
2. When the customization is ready for checkin, he clicks *Version Control > Pending objects > Check in*.
3. In the *Check in* form, Jens provides issue number and short description of the issue fixed:

Check in

Model elements

Description

GNN-1111 Sales order confirmation cannot be posted

* regression caused by the fix for GNN-1110. Fixed by adding the missing if-statement.

Code review

Type: Nice to have To: john.doe@cgi.com

Subject: Code review: GNN-1111 Sales order conf Cc:

From: jens.jensen@cgi.com

Include	Path	Version	Action
<input checked="" type="checkbox"/>	\Menus\DevelopmentTools	1	Edit

All None

OK Cancel

4. Jens clicks OK, so the change is checked in to the VCS, and an email is sent to developer John Doe. (As you can see, no additional actions were required from Jens to send the code for review)
5. John Doe gets a new email in his inbox. He opens the email.

GNN-1111 Sales order confirmation cannot be posted

☐ Jens Jensen <jens.jensen@cgi.com>

Sent: ti 11-06-2013 15:01

To: john.doe@cgi.com

Cc:

 Message  [5C2024F6-EEFE-4E3A-9D24-E135F946E14A].crp (27 KB)

GNN-1111 Sales order confirmation cannot be posted

* regression caused by the fix for GNN-1110. Fixed by adding the missing if-statement.

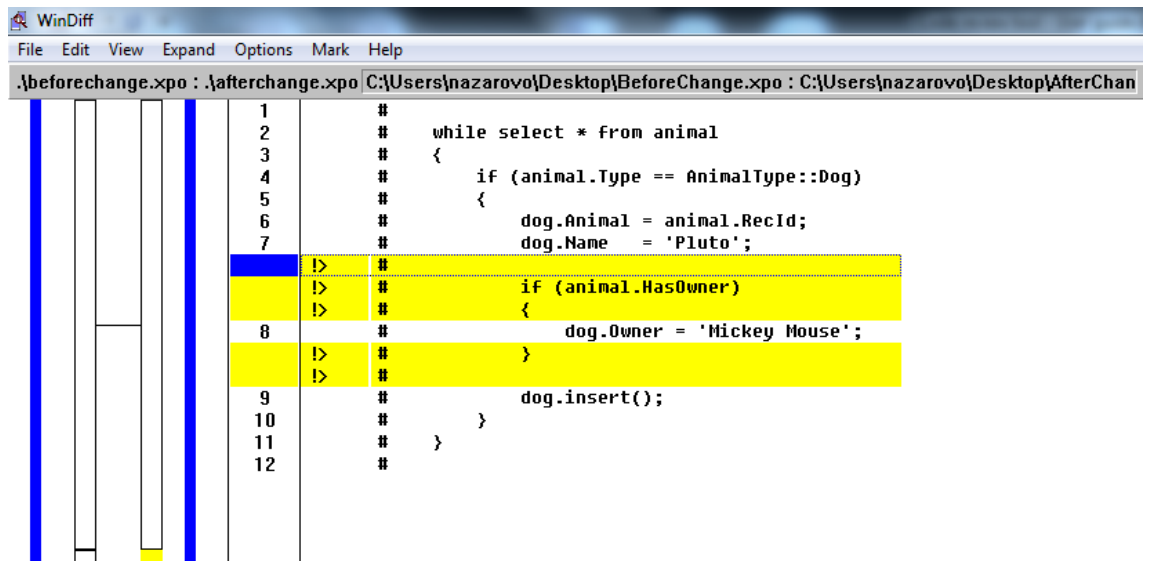
[Code review

[Type(Nice to have)]

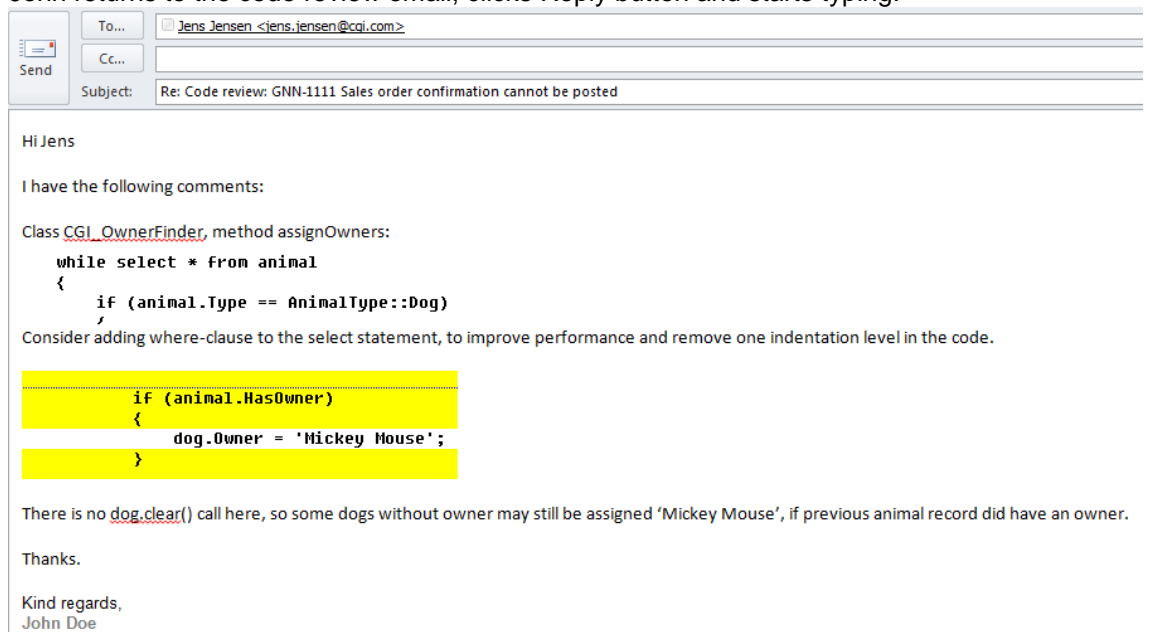
[ID({5C2024F6-EEFE-4E3A-9D24-E135F946E14A})]]

[Product name(Microsoft Dynamics AX 2012)]]

6. John reads the description and message attributes below the description, and double-clicks on the attachment:



7. There are a couple of issues in both the code, and John decides to send feedback to Jens.
8. John returns to the code review email, clicks Reply button and starts typing:



9. John clicks Send button, and the feedback is sent to Jens.
10. Jens receives the code review feedback.
11. He fixes both issues and replies to John with a simple *"Thank you. I have fixed the issues. Regards, Jens"*

4. Cookbook

4.1 You would like to have somebody review your code as soon as possible

Change code review type in the *Check-in* dialog to *Urgent* and select an explicit reviewer in the To field:

The screenshot shows a 'Review' window. Under the 'Model elements' tab, there's a 'Description' text area containing 'Sample change'. Below this, the 'Code review' section is highlighted with a green rectangle. It contains several fields: 'Type' is a dropdown menu currently showing 'Urgent'; 'To' is a text field with 'john.doe@cgi.com'; 'Subject' is 'Code review: Sample change'; and 'From' is 'jens.jensen@cgi.com'. At the bottom, there are tabs for 'Include', 'Path', 'Version', and 'Action', along with an 'All' button.

There are several other code review types available. Always check if you need to select something else than the default “Nice to have”.

Code review type	Description	When
Urgent	You would like somebody to review the code as soon as possible. For example, your change should be ready for a new build today.	Before checkin
Must have	You would like somebody to review the code, but this is not that urgent and may wait for a day or so.	Before or after checkin
Nice to have	(Default) You would not mind if somebody looked at your change and left feedback.	Before or after checkin
Not needed	You fixed a trivial best practice error or alike, so there is no risk to break anything. You would not want others to spend their time here.	After checkin
Already done	You have already received feedback on this change.	After checkin

4.2 You would like to send a pre-checkin code review

Open *Version Control > Pending Objects* form and press *Review* button.

The screenshot shows a 'Pending objects' window titled 'Version: 2, \Jobs\sendViaGmail'. It has two tabs: 'Overview' and 'General'. The 'General' tab is selected and shows a table with the following data:

Path	Version	Action
\Jobs\sendViaGmail	2	Edit

To the right of the table are four buttons: 'Refresh', 'Check in', 'Review (a)' (which is highlighted with a blue border), and 'Create project'.

A dialog will be open, similar to the one you use when checking in. Use description as a message to the reviewer:

Review

Model elements

Description

XYZ-1234 "Everything is broken"

Code review

Type: Must have To: john.doe@cgi.com

Subject: Code review: XYZ-1234 "Everything is broken" Cc:

From: jens.jensen@cgi.com

Include	Path	Version	Action
<input checked="" type="checkbox"/>	\\Jobs\\sendViaGmail	2	Edit

All None

OK Cancel

4.3 When checking-in last time, SMTP server failed, so no code review pack was sent. You would like to resend the change for review.

Open *Version Control > Changes* form. Select your change and press *Send for review* button.

Changes - Change number: 5637146590, 16-06-2013

Overview General

Change number	Status	Date	Time	User	Check-in description
5637146590	Checked in	16-06-2013	11:54:10	Admin	fixed issue with multi To- and Cc emails
5637146589	Checked in	16-06-2013	11:54:10	Admin	fixed issue with multi To- and Cc emails
5637146588	Checked in	16-06-2013	11:54:10	Admin	fixed issue with multi To- and Cc emails
5637146587	Checked in	16-06-2013	11:54:10	Admin	fixed issue with multi To- and Cc emails
5637146586	Checked in	16-06-2013	11:54:10	Admin	fixed issue with multi To- and Cc emails

Contents WinDiff Create project Send for review (a)

In the dialog, select parameters for code review email and press OK:

Microsoft Dynamics AX (1)

Send the change for code review

To: john.doe@cgi.com

Cc:

From: john.doe@cgi.com

Type: Must have

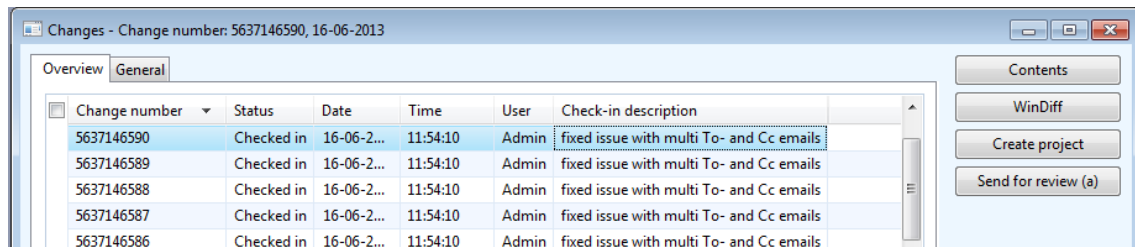
Subject: Code review: fixed issue with multi

OK Cancel

Code review type.

4.4 You would like to review changes done by one of your colleagues on the same project

Open *Version Control > Changes* form. Select a change and press *WinDiff* button.



4.5 You are working on your own dev box with AX 2012 at the customer network. You would like to avoid re-entering all email addresses to code review tool address book after each model store import or re-installation of AX

Check in the AX add-on Data Dictionary object (tables, EDTs and base enums prefixed with CgiCodeReview*) to the customer production code. Having these tables in production code will not affect business logic or data, but save you some time in the future. Also, this way you will avoid potential table ID conflicts when re-installing the modelstore on your development environment.

4.6 How to conduct a successful code review?

[Read](#) the article and comments on StackOverflow.com.

4.7 You received a code for review, and it looks like the developer has removed a lot of methods from a standard AX class

This is a known issue. That may happen when the Before version of the object comes from a lower layer, where other methods are defined, but not overridden in the upper layer. So, it looks like they are deleted in the After version. The code review tool is able to handle such a scenario when sending for pre-checkin review or checking in, but not when re-sending old change lists from the History. Look for tags that may indicate a change in that case.

4.8 Your team works on the same environment in AX 2009. You don't want the code review tool to be part of the XPO you deliver to customers.

Import the tool to another layer, say USR or VAP, and use it while developing in your actual development layer (e.g. VAR).

4.9 You got feedback in form of a text file, which contains lines numbers and comments

You have been sent a file that may be imported to *WinDiff* (see *File > Import Comments List...* in the menu), so the comments will appear right under corresponding changes. This is just another way to send feedback. You can either find the email thread with the original code review pack (if you remembered to Cc yourself) and double click it, or copy the file with comments to your development environment, find the changelist with the same code review ID in the version control history and click *WinDiff* button.

You can create the same sort of file when reviewing somebody's code. While in *WinDiff*, select the line that you would like to comment on and press *Alt+C*. This will open a dialog where you can write your comment. After you finished commenting, select *File > Save Comments List...* and create a text file. You can attach the file to the "feedback" email now.

4.10 What kind of feedback should be provided to code author?

Anything that may help the author improve his/her solution and/or make him/her learn something new. Regardless of whether you found a bug, a deviation from best practices or sub-optimal way of doing anything, let the author know what you think. Even if the author decided to keep his current solution as it is, (say, it is correct, but written in a way that you don't really like), he will get use of new knowledge in the future.