|  |
| --- |
| **NiSource** |
| **Columbia Pipeline Group (CPG)** |
| **ProTool Forecast Import/Export Excel Template**  **IT Support Manual** |

Document Version: 1.00

By: Michael Card, Centric Consulting

[1 Field Cost Tracking (FCT) Solution 3](#_Toc354819077)

[1.1 Solution Architecture Diagram 3](#_Toc354819078)

[1.2 Resource Information 4](#_Toc354819079)

[1.3 Forecast Import/Export – Excel Workbook 6](#_Toc354819080)

[2 Developers Reference 9](#_Toc354819081)

[2.1 Forecast Import/Export – Solution Design (ToDo: Rewrite all sections from this point for the Forecast Import/Export Solution. Current content was copied from the FCT Solution) 9](#_Toc354819082)

[2.2 Microsoft Related Documentation and Online Help 10](#_Toc354819083)

[2.3 Code Changes in the Excel Macro-Enabled Template 11](#_Toc354819084)

[2.4 Code Changes in the FieldCostTracker .NET VSTO Solution 12](#_Toc354819085)

[2.5 Building the FCT Excel Template from Visual Studio for Deployment 13](#_Toc354819086)

[2.6 Building the Setup.EXE Installer Package for Deployment 14](#_Toc354819087)

[3 FCT Workbook – Data Validation Errors 15](#_Toc354819088)

[4 FCT Workbook - Error Troubleshooting 17](#_Toc354819089)

[4.1 Summary of Error Codes 17](#_Toc354819090)

[4.2 Error Message Definitions 18](#_Toc354819091)

[5 FCT Setup - Error Troubleshooting 27](#_Toc354819092)

[5.1 Installation Error Definitions 27](#_Toc354819093)

[6 Revision History 34](#_Toc354819094)

# Field Cost Tracking (FCT) Solution

## Solution Architecture Diagram

## Resource Information

|  |  |
| --- | --- |
| Active Directory Group  &  User Access | **G\_MAXIMOUSERS**  Users must be a member of this group to be granted permissions to retrieve and save data to the ProTool database from Excel. The macros access data through the SQL Server ProTool database. Note: This is the infrastructure layer of security that enforces access to the SQL Server resource.  The permissions to work with Forecast data is enforced through the ProTool custom Web application user roles. Note: This is the application user/role security layer that enforces what tasks users can perform on the data. The Roles that determine what tasks users can perform inside the Excel Template is defined in the ProTool.Settings database table. |
| Forecast Excel Template Location  User’s  Setup | SharePoint URL: Ask Andy or Betsy?  Template File: Protool\_Forecast\_co\_proj\_year.xltm  Note: There is nothing to install on the users’ computers for this Template to function. |
| Maximo Web Service URL | Production URL: T.B.D. in Phase 2  Test URL: <http://swnhamapmaxt03a:7003/meaweb/services/NI_FCTTOMXREC_IF>  FCT Workbook Features: The “Submit to Maximo” button macro interfaces with this Web Service for creating service receipt transactions in Maximo.  Note: This URL is stored in the ProTool.Settings database table.  Tip: Web Service availability can be checked by manually navigating to the URL using the IE Browser. |
| ProTool Database Server | Production Server: CWNCHADBSQLP092\INST1  Database: ProTool  Connection String:  **Provider=SQLOLEDB.1;Data Source=BW2CHADBSQLT09\S2K8TST;Initial Catalog=ProTool\_Staging; Integrated Security=SSPI**  Test Server: BW2CHADBSQLT09\S2K8TST  Database: ProTool\_Staging  Connection String:  **Provider=SQLOLEDB.1;Data Source=BW2CHADBSQLT09\S2K8TST;Initial Catalog=ProTool\_Staging; Integrated Security=SSPI**  SQL Server Authentication: Windows Integrated via Active Directory Group  All macro buttons access this database for retrieving and saving Project forecast estimates.  Note: The ProTool database connection string is stored in the hidden DEBUG worksheet tab. This is not the best place because if the resource server name or SQL instance name ever changes pre-existing Templates would have to be changed or retired. For example: User’s will need to start new Workbooks from the updated Template and retire all their old Workbooks. |
| Source Code Location | GitHub Repositories: MCard/ProTool  Note: The Excel source Template, SQL deployment scripts, and other supporting documentation or files are currently stored at this location. |

## Forecast Import/Export – Excel Workbook

|  |  |
| --- | --- |
| **Template Feature** | **Description and Procedures** |
| Minimum Requriements | The Forecast Workbook has the follow minimum requirements in order to function successfully:   * Microsoft Office 2007 or 2010. Previous versions of Office are not supported and no guarantee the macros will function. * Minimum Hardware and Operating System requirements specified for installing and running MS Office 2007 or 2010. * The Workbook Macros require the computer be connected to the NiSource private network. |
| Workbook Version | The Workbook version identifies a revision of the VBA macro code that resides within each Workbook file.  To check the current version:   1. Open the Workbook in Excel. 2. Make sure the Forecast Information worksheet is in focus. 3. Place your mouse pointer over the top-right cell where the “Forecast Summary” title is displayed. This initially is cell G1.   Tip: Cross check the user’s Workbook version with the latest version of the Template in SharePoint to see if the user has the latest code. |
| Protected Worksheets | All of the worksheets within the Forecast Workbook are protected including the hidden DEBUG worksheet using the same password. Certain cells within each worksheet is unprotected to allow users to enter data.  Password: **fcast**  To unprotect a worksheet:   1. Set focus to the worksheet to be unprotected. 2. Select the “Review” menu ribbon. 3. Click on “Unprotect Sheet” in the Changes menu group. 4. Enter the password to unprotect the worksheet. |
| Protected VBA Code | The Workbook’s VBA macro code is hidden and protected with a different password than what is used on the protected worksheets.  Note: Code changes should only be made by a software developer and via a change request process.  Password: **wildcard**  To protect the VBA code:   1. Click the Developer menu ribbon and click Visual Basic to open the VBA project window. 2. In the VBA project window, click on Tools then VBAProject Properties. This opens the project properties dialog. 3. Click on the Protection tab. 4. Check mark the “Lock project for viewing”. 5. Enter the password twice and click OK. 6. Close the VBA project window. Next time the Template is opened and you try to view the VBA code a password will be required. |
| Hidden DEBUG Worksheet | The DEBUG worksheet tab is hidden and is the first tab of the workbook. It contains configuration settings needed by the VBA macros and additional details of any errors that might have occurred during macro processing.  Users should never unhide or make any changes to the DEBUG tab as part of their normal processes and procedures.  You can check for additional error detail by unhiding the DEBUG worksheet tab. Here you can obtain the error number; technical error message; and the message that was displayed to the user.  Technical error detail is not displayed to the users, but instead logged on the DEBUG tab.  Also, configuration settings can be modified if a data resource changes such as in the case of a network server change.  To unhide the DEBUG worksheet:   1. Right-click on any worksheet tab. 2. Select Unhide. 3. Select the DEBUG worksheet and click ok. 4. The DEBUG worksheet appears.   To change a configuration setting:   1. Unhide the DEBUG worksheet. 2. Unprotect the DEBUG worksheet. The password is the same as all other worksheets. See “Protected Worksheets” for procedures. 3. Make your change. 4. Enter numeric value of 7 in cell C1 and click the “Unit Test” button. This will re-hide the DEBUG worksheet and ensure all worksheets are re-protected. |
|  |  |
|  |  |

# Developers Reference

This section contains technical information and special procedures for software developers to reference during future development and maintenance.

## Forecast Import/Export – Solution Design (ToDo: Rewrite all sections from this point for the Forecast Import/Export Solution. Current content was copied from the FCT Solution)

The solution started out as a stand-alone Excel macro-enabled Template with only VBA code embedded in the Workbook. During development a decision was made that submitting receipt transactions to Maximo will occur in real-time by integrating with a new Apache AXIS Web Service. The Web Service is based on the SOAP protocol.

Consuming a Web Service directly from VBA code starting with MS Office 2007 is no longer supported by Microsoft. The new way for such advanced functionality is to build the solution as a Visual Studio Tools for Office (VSTO) project. This solution will not work on MS Office versions earlier than 2007.

As a result, the Field Cost Tracking solution is made up of both Excel VBA code and C# .NET code. All future development efforts must be performed using Visual Studio 2010 or newer.

The solution namespace to open from source control is: **FieldCostTracker.sln**

The VBA code is able to interface with VSTO code using COM Interop. Therefore, the FieldCostTracker VSTO project’s Build property, “Register for COM interop”, must be checked. The Setup.exe Windows installer registers the DLL component for COM on each user’s local computer.

The solution was intentionally targeted for .NET Framework 3.5 because this was the latest Framework already installed on users’ computers during the initial development according to the Enterprise Architecture (EA) team.

## Microsoft Related Documentation and Online Help

Here are some website links to online documentation and help with building and deploying VSTO solutions.

Architecture of Document-Level Customizations:

<http://msdn.microsoft.com/en-us/library/zcfbd2sk(v=vs.100).aspx>

Office Development with Visual Studio:

<http://blogs.msdn.com/b/vsto/archive/2010/01/21/vsto-2010-runtime-components-explained-aaron-cathcart.aspx>

Interop Between VBA and Visual Studio Office Solutions (VSTO):

<http://blogs.msdn.com/b/bethmassi/archive/2009/11/05/interop-between-vba-and-visual-studio-office-solutions-vsto.aspx>

Custom Document Properties Overview:

<http://msdn.microsoft.com/en-us/library/ds87aeyf(v=vs.100).aspx>

Mage.exe - Manifest Generation and Editing Tool:

<http://msdn.microsoft.com/en-us/library/acz3y3te.aspx>

Deploying a Visual Studio 2010 Tools for Office Solution using Windows Installer:

<http://msdn.microsoft.com/en-us/library/ff937654>

Forum discussion on where to deploy and share the VSTO components:

<http://stackoverflow.com/questions/2081040/best-content-on-how-to-deploy-and-share-a-vsto-solution>

## Code Changes in the Excel Macro-Enabled Template

IMPORTANT: Making changes to the Excel Template while running inside Visual Studio Debug mode will not be saved once the Template is closed and Debugging is stopped.

Follow these exact procedures for making any changes to the Excel Template including changes to VBA code, worksheet layouts, formatting, or any other content.

1. Get latest solution files from source control. This includes the Excel template and worksheet objects.
2. Open the FieldCostTracker.sln solution in Visual Studio 2010.
3. Do a source check-out on the CostTrack\_proj\_po\_template.xltm file.
4. Use Windows Explorer and navigate to the local source folder where the template file exists.
5. Double-click the template file to open it in MS Excel.
6. Make whatever changes are required.
7. Remember: Update the version number on the hidden Debug tab and on the PO Detail cell with the comment. Use the Unit Test macro button to unprotect and re-protect all worksheets in one shot.
8. Click Save As and select file type Macro-Enabled Template. Navigate back to the source folder and select the same template file to overwrite it with your new changes.
9. You can test your changes before closing, or test using Visual Studio Debug. If you want to step into the VSTO code you must unit test using Visual Studio.
10. Tip: Don’t save the Template after testing, so you don’t leave test data in the Template file.
11. If you need to make a correction, exit Debug mode; answer No to save template (test data) changes; and repeat steps 3 through 9 above.
12. Follow instructions later in this document for building and deploying an updated FCT Macro-Enabled Excel Template.

## Code Changes in the FieldCostTracker .NET VSTO Solution

Note: Any change at all in the FieldCostTracker assembly (DLL source) will require building a new Setup.exe installer with a newer version. All users will need to run the new Setup.exe in order to deploy the new DLL to each user’s computer.

Follow these exact procedures for making any changes to the FieldCostTracker DLL component.

1. Get latest source code from source control.
2. Increment the assembly version number. In the project properties, Appication tab, click the Assembly Information button. Increment and keep both the Assembly Version and File Version the same.
3. Increment the version number on the DLL and deployment manifest files with the exact same version number from previous step.
4. In the project properties, Build Events tab, you will see two Post-build event command lines that are shown below:

**Important: Visual Studio was not incrementing these versions on every build, so these post-build commands are very critical for the success of upgrading user computers without errors.**

"C:\Program Files\Microsoft SDKs\Windows\v7.0A\bin\Mage.exe" -u $(TargetName)$(TargetExt).manifest -v 1.0.1.9 -cf FieldCostTracker\_TemporaryKey.pfx

"C:\Program Files\Microsoft SDKs\Windows\v7.0A\bin\Mage.exe" -u FieldCostTracker.vsto -v 1.0.1.9 -appm $(TargetName)$(TargetExt).manifest -cf FieldCostTracker\_TemporaryKey.pfx -pub "NiSource NiPADS"

1. Find and change the version number on both lines, and click the save icon in the toolbar.
2. Make whatever changes are needed and unit test the changes by running in Debug mode.
3. If you changed the interface signature of the ICostTrackManager or IPoDetailSheet such as adding a method or adding/changing an argument of an existing method, perform these steps as well:
4. Rebuild the FieldCostTracker project using the Debug build configuration. This will re-register the COM Interop on the local dev computer.
5. Follow the previous section’s instructions for changes to the Template, but instead of a code change, you will need to refresh the Reference to the FieldCostTracker DLL.
6. Refresh by selecting Tools -> References and uncheck the FieldCostTracker reference and click OK to close the dialog.
7. Add the reference back by selecting Tools -> References, then find and check the FieldCostTracker reference and click OK to keep the reference.
8. After saving the change to the same macro-enabled Template, be sure to perform a test to ensure the reference to the modified DLL is functional.
9. Rebuild the project using the Debug build configuration.
10. Follow instructions later in this document for building and deploying an updated FCT Setup Installer.
11. Check all changes back into source control with comments of what changed.

## Building the FCT Excel Template from Visual Studio for Deployment

You must follow these instructions to create a new Excel Template if any change was made to the Template content or its VBA code including refreshing the FieldCostTracker DLL reference.

1. In the Visual Studio FieldCostTracker project view the ThisWorkbook.cs code.
2. VSTO automatically creates a custom property called \_AssemblyLocation embedded in the Workbook that tells Excel where to find the installed FieldCostTracker.dll component. The issue with this is that VS sets the property to the local source code bin path instead of the user’s program files path. The code between the #if (DEBUG) directives in ThisWorkbook will overwrite this property to the correct deployment path. The remainder of these instructions also exist in the comments at the top of the ThisWorkbook partial class.
3. Uncomment the appropriate Const called DEPLOYMENT\_MANIFEST\_FILE that contains the path and file name where the FieldCostTracker.vsto deployment manifest file will be installed on the end-user's computers. Notice there is one for a 32bit and another one for targeting a 64bit installation path.
4. Important: Un-install the FieldCostTracker document-level customizations from Programs & Features if it exists before the next step. This is the one Excel installs, not the one that the Setup.exe created. You can tell the Setup.exe because it’s version format is like this: 1.0.1009. The Excel customization version format is like this: 1.0.1.9.
5. Run the solution in Debug mode.
6. At this point the Template will open and the \_AssemblyLocation custom property will be set to the correct user target path.
7. Perform instructions to protect all the worksheets (if not already protected) and protect the VBA source code using the separate password as described earlier in this document.
8. Do a Save As and select macro-enabled template for the file type. Save the template to a local folder.
9. Close the Excel Template to exit Debug run mode.
10. Copy the saved Template from your local path to the SharePoint location where users can open it.
11. Repeat these steps 1-7 for creating a 2nd Template if needed that finds the deployment manifest when used on a 64 bit computer.

## Building the Setup.EXE Installer Package for Deployment

You must follow these instructions to create a new Setup.exe Installer package if any change was made to the VSTO project source code. This will re-build the FieldCostTracker.dll component and all it’s dependencies.

1. Switch to the **Release** build configuration in Visual Studio.
2. Increment the assembly version of the FieldCostTracker project if not already performed previously during code changes. Right-click on the project and select project properties. From the Application tab, click on Assembly Information.
3. Increment the version in the post-build events command lines using the same version if not already performed previously as described in the code change procedures.
4. Increment the FieldCostTracker.Setup version. Left click on the Setup project and open the Properties view. Modify the Version property and answer YES to generate a new GUID.
5. Check the ProToolConnectionString setting value under the project properties, Resources tab. Modify it as needed so it connects to the appropriate SQL Server and database based on the target deployment environment.
6. Rebuild the FieldCostTracker project.
7. Right-click the “Detect Dependencies” folder under the FieldCostTracker.Setup project, and click Refresh Dependencies.
8. Exclude any duplicate dependencies that might show up like the FieldCostTracker.tlb file. Right-click each one and check “Exclude”. A red circle will appear indicating it will be excluded from the deployment.
9. Rebuild the FieldCostTracker.Setup project.
10. Create a new empty folder somewhere on your local drive called “FCT Install”
11. Navigate to the Setup project’s Release folder using Windows Explorer.
12. Copy all contents from the Release folder to the new FCT Install folder including the Setup.Exe, MSI, and subfolder.
13. Use WinZip to zip the new FCT Install folder and rename the zip file for whatever the target environment is such as FCT Installer TEST.zip or FCT Installer PROD.zip.
14. Repeat steps 1 through 12 to build a separate FCT Installer for each targeted environment.

# FCT Workbook – Data Validation Errors

Below is a sample list of various data validation error messages that could be presented to users when performing the different processes with the FCT Workbook.

1. "P.O. Number cannot be more than 8 characters long."
2. "P.O. Number is required."
3. "Purchase order details could not be found for P.O.# '{0}'." -- User should recheck the PO number entered and verify it exists in Maximo. Also, if the PO was created in the same day it won’t be available to pull into the FCT Workbook until next day.
4. "Could not load purchase order '{0}' because it has a status of '{1}'." – The purchase order has to be in the “SENT” status. Only sent purchase orders can be retrieved into the FCT Workbook.
5. "You have not 'submitted' the previous period ending {0}."

"This is likely because you have manually entered the receipts using the Maximo application, or you are not ready to submit yet."

"Do you want to disable the submit feature for period ending {0} ?"

"Note: If you answer 'yes' you will need to manually create receipts using the Maximo application for that period." -- This is asking the user if he/she wants to disable the “Submit to Maximo” function for the previous period that they are copying since they are creating a new worksheet for a new period. It is advisable to disable the submit if they have already or are planning to enter the receipts manually via the Maximo application.

1. "The submit feature is disabled for this period worksheet. Please use the Maximo application to create receipts." -- The “Submit to Maximo” was automatically disabled during the “Copy to New Sheet” button click without prompting the user. This is because the Maximo Web Service has not yet been deployed, and the user has no choice except entering the service transaction receipts directly through the Maximo application.
2. "Please contact IT Support for assistance if you continue to get this error." – This sentence is appended to various internal error messages that appear. See Troubleshooting section later in this document.
3. "Connecting to the NiSource server failed!"

"Please make sure you are logged into the NiSource network, and try your task again." – Verify the user is properly connected to the NiSource private network. If connected, seek support from the ProTool support team.

1. "Access to P.O. information was denied. You might not be a member of the security group 'G\_MAXIMOUSERS'." -- Check if the user is a member of the Active Directory group. If the user has authorization, add them to the Group. If not authorized, instruct user to follow standard process for requesting authorization.
2. "The following error occurred that indicates the workbook layout needs to be corrected. Please contact IT Support for assistance in fixing this Excel workbook." – This error will only appear if somehow a user did something in Excel that caused one or more of the worksheets to become corrupt. The VBA code locates specific rows and cells by searching for key words or characters (e.g. “\*”, “\*\*”, etc.) in specific columns that are protected. Users can not modify these protected cells and columns, nor can they delete them. However, if they found a way such as deleting the entire worksheet or unprotecting the worksheet with the secret password the VBA code will throw this error if the layout is not in the expected state. A programmer could fix the issue by correcting the layout or replacing the key values. Alternatively, the user could start over with a new Workbook from the default Template.
3. "A program error has occurred. You can try your task again." – This is the default, user-friendly message whenever an internal program error occurs. You can view the technical detail message in the hidden Debug tab. See Troubleshooting section later in this document for various technical error messages that can occur.

**Errors Related to Submit to Maximo:**

1. "An internal program error was returned from Maximo. You can try your task again." – This is the default, user-friendly message whenever an internal program error occurs inside the Maximo Web Service. You can view the technical detail message in the hidden Debug tab. Seek level 3 support from the Maximo team and share the technical detail for resolving errors that occur in the Maximo Web Service.
2. "Maximo returned the following validation error. No receipt transactions were created in Maximo." Note: various validation error messages returned by the Maximo Web Service will be appended to this message once phase 2 is complete and rolled out. This is usually the result of some invalid state of either the submitted data or the related data in Maximo. Seek support from the Maximo support team if the message returned from Maximo is unclear.
3. "Maximo returned a validation error, but an error occurred while trying to extract the message. No receipt transactions were created in Maximo." – Some invalid data error occurred in the Maximo Web Service, but there was also a technical error trying to extract the message from the SOAP Response returned from the Web Service. This could be the result of the response format changing or some technical failure inside the Maximo Web Service. Seek support from the Maximo support team and share the technical detail from the hidden Debug tab if this ever occurs.
4. "Connecting to the Maximo server failed!"

"Please make sure you are logged into the NiSource network, and try your task again." -- If the user is connected to the NiSource private network, then check the server state and seek network and/or Maximo support.

1. "The Maximo service is currently not available or could not be found. You can try your task again later." – The Maximo Web Service might be down. Seek network and/or Maximo support.
2. "The 'Send to Maximo' feature is not yet avialable. Alternatively, you can use Maximo to enter your receipt transactions manually."

# FCT Workbook - Error Troubleshooting

This section shows samples of various errors that could occur while users are working in the Excel Workbook. A definition of each error along with the recommended troubleshooting steps to resolve each issue is described throughout the sub-sections below.

If a user reports an error you can do a Search on this section using key words from the displayed message, and find the corresponding error sample below for troubleshooting.

## Summary of Error Codes

**Microsoft VBA Error Codes:**

* < 1000 - VBA internal error code. Check technical detail on DEBUG worksheet tab for more information.
* -2147217843 - Authentication to the SQL Server ProTool database failed. Check if user is a member of the appropriate Active Directory group.
* -2147467259 - Connectivity to the SQL Server failed or server not found. Check if user is logged into NiSource private network (VPN).

**FCT Custom Error Code Definitions:**

* 1000-1999 - Represents data validation errors generated from this VBA code.
* 2000-2899 - Represents data validation errors returned from the Maximo Web Service.
* 2900 - Connectivity to the Maximo Web Service failed. Check if user is logged into the VPN.
* 2910 - Found Web Server, but the Maximo Web Service is unavailable or not found. Check if Service is down or wrong URL address.
* 2990 - The Invalid Data message from the Maximo Web Service SOAP response is malformed and could not be extracted. Check recent Maximo code changes for bugs.
* 2999 - An internal Exception occurred in the Maximo Web Service. Check recent Maximo code changes for bugs.
* 9000 - Worksheet layout is in an unexpected state. The VBA code could not find a protected cell value. Obtain a copy of the Excel file and correct the layout or replace expected missing values.
* 9010 - A function's parameter was not properly set to an instance of an object. This is a programmer's bug.
* 9020 - A configuration setting name could not be found on the DEBUG worksheet tab. Either the sheet layout changed or the setting is missing.
* 9999 - An internal Exception occurred inside the Visual Studio Tools for Office (VSTO) C# .NET code. See DEBUG tab for technical details.

## Error Message Definitions

**Error Number:** -2147467259

**Error Source:**  FCT Workbook / SQL Server

**Process:**  Get PO Detail

**User Displayed Message:**

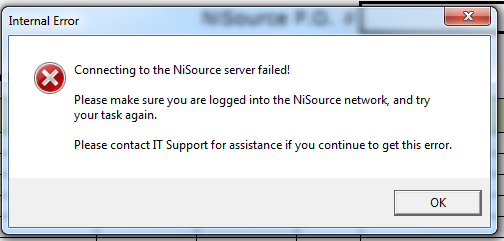
Connecting to the NiSource server failed!Please make sure you are logged into the NiSource network, and try your task again.Please contact IT Support for assistance if you continue to get this error.

**Technical Error Message:**

[DBNETLIB][ConnectionOpen (Connect()).]SQL Server does not exist or access denied.

**Possible Causes:**

The SQL Server where the ProTool database might be down, or the connection string in the configuration settings on the hidden DEBUG worksheet tab is no longer valid.



**Error Number:** 2000-2899

**Error Source:**  Maximo Web Service

**Process:**  Send to Maximo

**User Displayed Message:**

Maximo returned the following validation error. No receipt transactions were created in Maximo.PONUM is not valid.

**Technical Error Message:**

System.Web.Services.Protocols.SoapException: nisource.iface.migexits.NiFieldCostTrackingToMaximoReceipts$NiValidationException: Data validation failed for inbound transaction. Errors are: [message]10|PONUM is not valid.[/message].

at System.Web.Services.Protocols.SoapHttpClientProtocol.ReadResponse(SoapClientMessage message, WebResponse response, Stream responseStream, Boolean asyncCall)

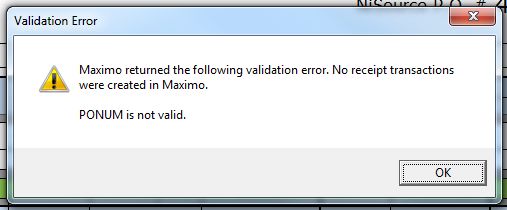
at System.Web.Services.Protocols.SoapHttpClientProtocol.Invoke(String methodName, Object[] parameters)

at FieldCostTracker.MaximoServiceProxy.NI\_FCTTOMXREC\_IF.processDocument(NI\_FCTTOMXREC\_IF1 NI\_FCTTOMXREC\_IF) in C:\Projects\NiSource\TFS\NiSource - Field Cost Tracking\FieldCostTracker\Web References\MaximoServiceProxy\Reference.cs:line 81

at FieldCostTracker.CostTrackManager.SubmitPoReceipts(String userName, String password, String poNumber, String costSheetName, String poRows, String receiptRows, String receiptMemo) in C:\Projects\NiSource\TFS\NiSource - Field Cost Tracking\FieldCostTracker\CostTrackManager.cs:line 183

**Possible Causes:**

This is just one of the many data validation errors that can occur in Maximo. This is normal behavior and not a program bug. In this example, ask the user to check the PO status in Maximo.



**Error Number:** 2900

**Error Source:**  FieldCostTracker .NET VSTO Assembly

**Process:**  Send to Maximo

**User Displayed Message:**

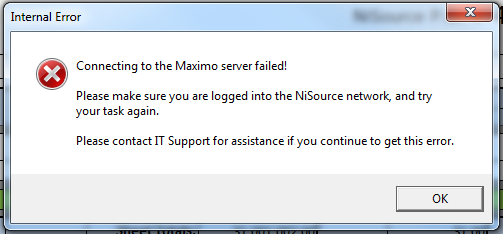
Connecting to the Maximo server failed!Please make sure you are logged into the NiSource network, and try your task again.Please contact IT Support for assistance if you continue to get this error.

**Technical Error Message:**

System.Net.WebException: The remote name could not be resolved: 'swnhamapmaxt03a'\r\n at System.Net.HttpWebRequest.GetRequestStream(TransportContext& context)\r\n at System.Net.HttpWebRequest.GetRequestStream()\r\n at System.Web.Services.Protocols.SoapHttpClientProtocol.Invoke(String methodName, Object[] parameters)\r\n at FieldCostTracker.MaximoServiceProxy.NI\_FCTTOMXREC\_IF.processDocument(NI\_FCTTOMXREC\_IF1 NI\_FCTTOMXREC\_IF) in C:\\Projects\\NiSource\\TFS\\NiSource - Field Cost Tracking\\FieldCostTracker\\Web References\\MaximoServiceProxy\\Reference.cs:line 81\r\n at FieldCostTracker.CostTrackManager.SubmitPoReceipts(String userName, String password, String poNumber, String costSheetName, String poRows, String receiptRows, String receiptMemo) in C:\\Projects\\NiSource\\TFS\\NiSource - Field Cost Tracking\\FieldCostTracker\\CostTrackManager.cs:line 182

**Possible Causes:**

The user’s VPN is disconnected; or the Server hosting the Service is down. It is possible the URL address is wrong, but this is unlikely if the submit to Maximo feature has been working for the current Excel workbook.



**Error Number:** 2910

**Error Source:**  Maximo Web Service

**Process:**  Send to Maximo

**User Displayed Message:**

The Maximo service is currently not available or could not be found. You can try your task again later. Please contact IT Support for assistance if you continue to get this error.

**User Displayed Message (before 4/15/2013):**

The 'Send to Maximo' feature is not yet avialable. Alternatively, you can use Maximo to enter your receipt transactions manually. Please contact IT Support for assistance if you continue to get this error.

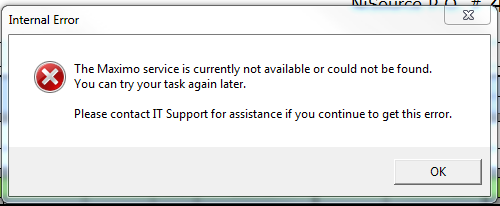
**Technical Error Message:**

System.Web.Services.Protocols.SoapException: The AXIS engine could not find a target service to invoke! targetService is NI\_FCTTOMXREC\_IF\r\n at System.Web.Services.Protocols.SoapHttpClientProtocol.ReadResponse(SoapClientMessage message, WebResponse response, Stream responseStream, Boolean asyncCall)\r\n at System.Web.Services.Protocols.SoapHttpClientProtocol.Invoke(String methodName, Object[] parameters)\r\n at FieldCostTracker.MaximoServiceProxy.NI\_FCTTOMXREC\_IF.processDocument(NI\_FCTTOMXREC\_IF1 NI\_FCTTOMXREC\_IF) in C:\\Projects\\NiSource\\TFS\\NiSource - Field Cost Tracking\\FieldCostTracker\\Web References\\MaximoServiceProxy\\Reference.cs:line 81\r\n at FieldCostTracker.CostTrackManager.SubmitPoReceipts(String userName, String password, String poNumber, String costSheetName, String poRows, String receiptRows, String receiptMemo) in C:\\Projects\\NiSource\\TFS\\NiSource - Field Cost Tracking\\FieldCostTracker\\CostTrackManager.cs:line 182

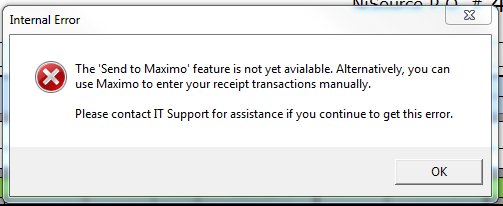
**Possible Causes:**

The URL Address is wrong or the Service is not operational on the server. If this error occurs before 4/15/2013, it is possible the Maximo Web Service has not been deployed into production yet.

On or After 4/15/2013:



Before 4/15/2013:



**Error Number:** 2990

**Error Source:**  Maximo Web Service

**Process:**  Send to Maximo

**User Displayed Message:**

Maximo returned a validation error, but an error occurred while trying to extract the message. No receipt transactions were created in Maximo.Please contact IT Support for assistance if you continue to get this error.

**Technical Error Message:**

The data validation message embeded in the SOAP response from the Maximo Web Service is malformed. Was expecting 2 array elements, but only found 1 elements. Here is the SOAP response: System.Web.Services.Protocols.SoapException: nisource.iface.migexits.NiFieldCostTrackingToMaximoReceipts$NiValidationException: Data validation failed for inbound transaction. Errors are: [message]PONUM is not valid Error code 10.[/message]. at System.Web.Services.Protocols.SoapHttpClientProtocol.ReadResponse(SoapClientMessage message, WebResponse response, Stream responseStream, Boolean asyncCall)

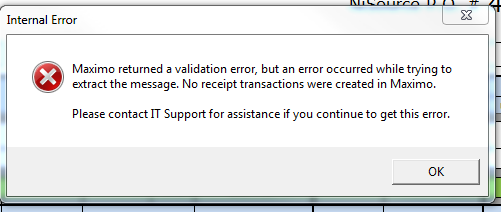
at System.Web.Services.Protocols.SoapHttpClientProtocol.Invoke(String methodName, Object[] parameters) at FieldCostTracker.MaximoServiceProxy.NI\_FCTTOMXREC\_IF.processDocument(NI\_FCTTOMXREC\_IF1 NI\_FCTTOMXREC\_IF) in C:\Projects\NiSource\TFS\NiSource - Field Cost Tracking\FieldCostTracker\Web References\MaximoServiceProxy\Reference.cs:line 81 at FieldCostTracker.CostTrackManager.SubmitPoReceipts(String userName, String password, String poNumber, String costSheetName, String poRows, String receiptRows, String receiptMemo) in C:\Projects\NiSource\TFS\NiSource - Field Cost Tracking\FieldCostTracker\CostTrackManager.cs:line 183

**Possible Causes:**

A program change may have occurred in the Maximo Web Service that changed the expected format of the return message embedded inside the Soap Response. The FCT solution uses a Regular Expression to extract and parse the message content between these strings [message] [/message].

It the case of this sample Soap Response above the highlighted string should of looked like this:

[message]10|PONUM is not valid.[/message]



**Error Number:** 2999

**Error Source:**  Maximo Web Service

**User Displayed Message:**

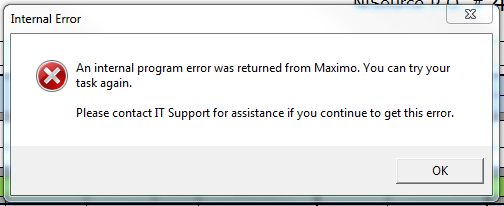
An internal program error was returned from Maximo. You can try your task again.Please contact IT Support for assistance if you continue to get this error.

**Technical Error Message:**

System.Web.Services.Protocols.SoapException: psdi.util.MXApplicationException: iface#STOP\_TRANSACTION at System.Web.Services.Protocols.SoapHttpClientProtocol.ReadResponse(SoapClientMessage message, WebResponse response, Stream responseStream, Boolean asyncCall) at System.Web.Services.Protocols.SoapHttpClientProtocol.Invoke(String methodName, Object[] parameters) at FieldCostTracker.MaximoServiceProxy.NI\_FCTTOMXREC\_IF.processDocument(NI\_FCTTOMXREC\_IF1 NI\_FCTTOMXREC\_IF) in C:\Projects\NiSource\TFS\NiSource - Field Cost Tracking\FieldCostTracker\Web References\MaximoServiceProxy\Reference.cs:line 81 at FieldCostTracker.CostTrackManager.SubmitPoReceipts(String userName, String password, String poNumber, String costSheetName, String poRows, String receiptRows, String receiptMemo) in C:\Projects\NiSource\TFS\NiSource - Field Cost Tracking\FieldCostTracker\CostTrackManager.cs:line 184

**Possible Causes:**

This can be any unknown program error. Most likely it is the result of a bug and requires debugging by a software developer for the Maximo Web Service and possibly the FieldCostTracker .NET VSTO solution as well. In the highlighted example above, it was discovered that this was a result of not including a required field (i.e. <SiteID>) in the Web Request XML.



**Error Number:** 9030

**Error Source:**  FCT Workbook – VBA code

**User Displayed Message:**

A program error has occurred. Microsoft Excel might have an old version of the FieldCostTracker customization installed.Please contact IT Support for assistance with uninstalling the older version.

**Technical Error Message:**

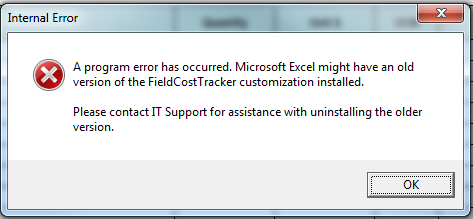
Error occurred trying to obtain an instance of the CostTrackManager from the FieldCostTracker.dll component. MS Excel probably has an old version of the FieldCostTracker customization already installed. Try closing Excel and uninstall the first FieldCostTracker that has a version format #.#.#.# (3 dots) using Control Panels. Then, reopen the Workbook and Excel will install the newer component.The Inner Exception: Object variable or With block variable not set

**Possible Causes:**

Most likely a new version of the FieldCostTracker DLL component was recently installed along with a new version of the FCT Excel Template. The user might not of uninstalled the previous versions of the FieldCostTracker from Add/Remove programs. It is also possible the user did not run the latest Setup.exe to installed DLL, but is attempting to use the new Excel Template. Perform these steps to resolve this issue:

1. Make note of the current Workbook version that was used during the error.
2. Close all instances of MS Excel.
3. Go to Control Panels -> Add/Remove Programs or Uninstall Programs (link name varies depending on which version of Windows and Control Panel view).
4. Look for 2 instances of the “FieldCostTracker”. The first instance is usually the customization that Excel installed and has a version format of #.#.#.# (3 dots). Compare this version with the version of the Workbook used. If the Workbook has a newer version this confirms the root cause of the error.
5. Regardless of the version, uninstall the first FieldCostTracker.
6. The 2nd FieldCostTracker instance with a version format of #.#.#### (2 dots) is from the DLL and dependencies being installed when the Setup.exe program was installed. If this instance version resembles the latest Workbook version, do not uninstall it—user already ran the latest Setup.exe. For example, Workbook version 1.0.2.1 and FieldCostTracker version 1.0.2001 is the correct installation.
7. If the version of the 2nd FieldCostTracker instance is older, then uninstall it too. Then, follow instructions to install the latest FCT Installer package (Zip file) to re-install the DLL and any dependencies.
8. Re-open the Workbook and try the user’s task again. You should notice MS Excel will install customizations the first time opening the file. Answer Yes to accept the install.

See screen shot on next page.



# FCT Setup - Error Troubleshooting

This section shows sample screen shots of technical errors that were experienced during development and testing of the FCT Setup.exe installer. If any of these errors occur it will most likely be the result of some hardware or software infrastructure change to the end-user’s environment such as an OS or MS Office upgrade.

These types of errors will most likely require a level 3 support developer to resolve the issue.

## Installation Error Definitions

**Process:**  Opening the FCT Excel Template.

**Technical Error Message:**

Installing Office customization – There was an error during installation.

System.Deployment.Application.DeploymentDownloadException: Downloading file: ///C:/Program Files/NiSource/FieldCostTracker/FieldCostTracker.vsto did not succeed.

System.Net.WebException: Could not find a part of the path ‘C:\Program Files\NiSource\FieldCostTracker\FieldCostTracker.vsto’.

**Possible Causes:**

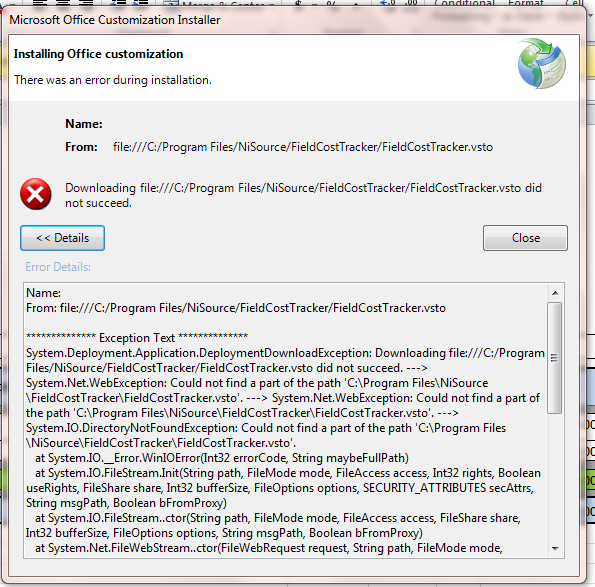
1. The user might not have installed the FCT Setup.exe (FieldCostTracker VSTO component). This component must be installed for the “Submit to Maximo” button macro to function. Other button macros might be functional if the error is ignored.

**Resolution:** Ask the user if they downloaded; unzipped; and ran the FCT Setup.exe, or check if the C:\Program Files\NiSource\FieldCostTracker folder exists with files. If not installed, see the user training guide to install the components.

1. The path location of where the FCT Setup.exe (FieldCostTracker VSTO components) have installed might be different from where the FCT Template is expecting them to be.

**Resolution:** Use “Add/Remove Programs” feature of Windows Control Panels to uninstall the FieldCostTracker components. Run the FCT Setup.exe to re-install the components to the “default” folder path.

**Screen Shot:**



**Process:**  Opening the FCT Excel Template.

**Technical Error Message:**

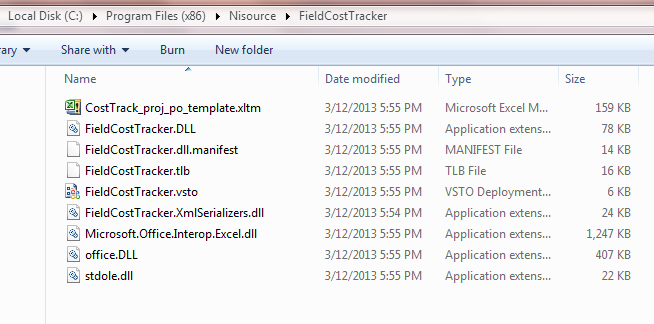
Could not load file or assembly ‘Microsoft.Office.Interop.Excel.Version=12/14.

Or one of its dependencies. The system cannot find the file specified.

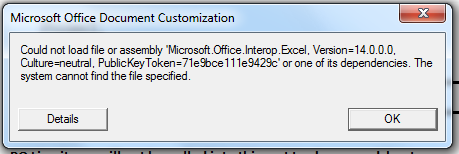
**Possible Causes:**

1. One of the FieldCostTracker VSTO component files might not have been installed with the FCT Setup.exe installation. The solution is dependent on certain component files being installed on the user’s computer.

**Resolution:** Try reinstalling or repairing the FieldCostTracker installation. Compare the files installed on the user’s computer with the image below that shows what files should exist. If reinstalling does not resolve, a programmer may need to check the Setup.exe solution for issues. Note: The Excel template with extension xltm is not required, but gets copied to the program files folder anyway.



**Screen Shot:**



**Process:**  Opening the FCT Excel Template.

**Technical Error Message:**

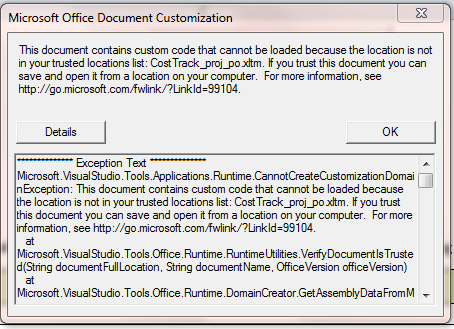
This document contains custom code that cannot be loaded because the location is not in your trusted locations list: CostTrack\_proj\_po.xltm. If you trust this document you can save and open it from a location on your computer.

**Possible Causes:**

1. The user might be trying to open the FCT Template using MS Excel 2007, and the template is located on a network share; in SharePoint; or in another trusted location. MS Excel 2010 seems to handle this better by asking the user if he/she wants to trust the file.

**Resolution:** Try downloading or copying the Template to the user’s local Documents folder and then open the Template. If this does not work, the user’s Trust settings in MS Excel might need to be looked at to determine why it’s not trusting the path and file.

**Screen Shot:**



**Technical Error Message:**

The customizations cannot be installed because another version is currently installed and cannot be upgraded from this location. To install this version of the customization, first use Add or Remove Programs to uninstall this program: FieldCostTracker. Then install the new customization from the following location.

**Possible Causes:**

1. Most likely a new version of the FieldCostTracker DLL component was recently installed along with a new version of the FCT Excel Template. The user might not of uninstalled the previous versions of the FieldCostTracker from Add/Remove programs.

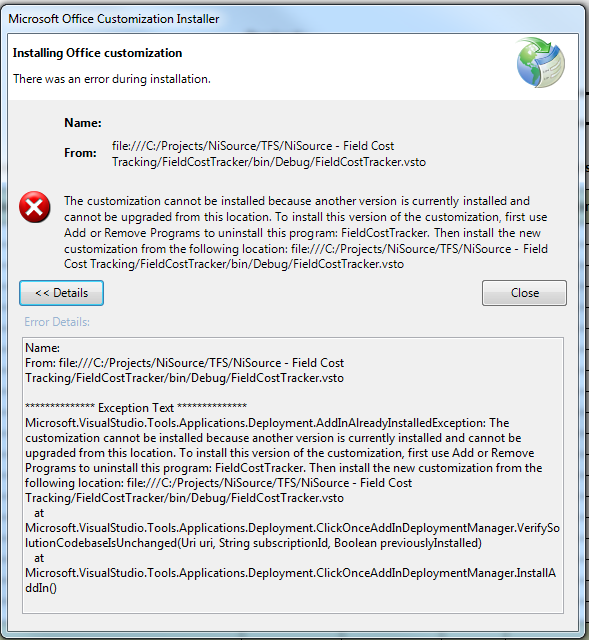
**Resolution:** Perform these steps to resolve this issue:

1. Close all instances of MS Excel.
2. Go to Control Panels -> Add/Remove Programs or Uninstall Programs (link name varies depending on which version of Windows and Control Panel view).
3. Look for 2 instances of the “FieldCostTracker”. The first instance is usually the customization that Excel installed and has a version format of #.#.#.# (3 dots).
4. Uninstall the first FieldCostTracker instance with version format #.#.#.#.
5. Re-open the Workbook and MS Excel will install the new customizations. Answer Yes to accept the install if prompted.

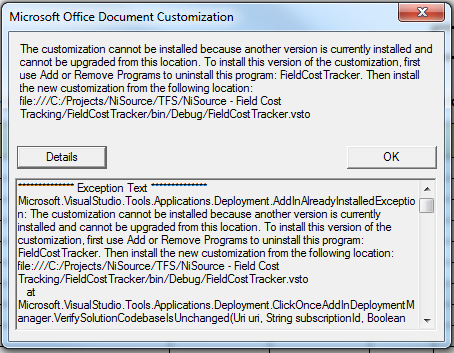
**Screen Shot:**

See screen shots on next two pages related to this error.

Screen 1 of 2 that appear for this error:



Screen 2 of 2 that appear for this error:



# Revision History

Document Owners(s):

| **Name** | **Role** |
| --- | --- |
|  |  |
|  |  |
|  |  |

The version history for this document:

| **Version Number** | **Date Issued** | **Issued By** | **Section Modified** | **Summary of Amendments** |
| --- | --- | --- | --- | --- |
| 1.00 |  | M.Card | All | Initial Draft |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |