

Microsoft Dynamics® GP

Recording Test Cases with the Macro System

White Paper

This paper describes how to use the macro system in the
Dexterity runtime to record test cases for use in regression
testing.

Date: August, 2008



Table of Contents

Introduction.....	3
Starting Data	3
Setting up the Microsoft Dynamics GP Environment.....	3
Macro Folder	3
Applications	4
System Settings	4
Dex.ini File.....	4
Keys.ini File	5
Recording a Test Case.....	6
Tips When Recording Test Cases.....	8
Comments	8
Fields	8
Windows.....	8
Scrolling Windows.....	8
Field Dumps.....	8
Reports	9
Application Problems When Recording Test Cases	9
Incomplete Functionality.....	9
Bugs in the Application	9
Running a Regression	9
Verifying the Test Case Results.....	10
Maintaining Test Case Macros	10
Additional Resources	10
Conclusion	10

Introduction

Testing is one of the most important steps in developing an application that integrates with Microsoft Dynamics GP. This document explains how to use the macro system provided by the Dexterity runtime to record test cases for your integrating application. After the test cases have been recorded, they can be used for regression testing. In regression testing, the macros are played back on future builds of your integrating application to verify that it is still functioning properly.

This document does not explain how to write test cases. Instead, it focuses on how to use the macro system to record test cases so that regression testing can be automated. Topics covered include:

- [Starting Data](#)
- [Setting up the Microsoft Dynamics GP Environment](#)
- [Recording a Test Case](#)
- [Tips When Recording Test Cases](#)
- [Application Problems When Recording Test Cases](#)
- [Running a Regression](#)
- [Verifying the Test Case Results](#)
- [Maintaining Test Case Macros](#)
- [Additional Resources](#)

Starting Data

When recording and playing back a test case, it is necessary to have a consistent starting point. The macro must have the same starting data every time it is run. There are many options for creating starting data, but the recommended option is to create starting data macros. These macros set up the starting data, and are run before the test cases are recorded or played back. Use the following guidelines when creating the starting data for your test cases:

- Install all of the applications that will be needed for you test cases.
- Think about all of the tests that will be automated. Can one set of starting data be used for all of the tests, or would multiple sets of starting data be more useful?
- When creating starting data, make the data as realistic as possible. Use a variety of settings instead of simply accepting defaults. Set up all item types, document types, and so on. However, avoid including complex "edge case" scenarios that only a small fraction of customers may be using. Instead, put these edge case tests into their own test case macros.

When creating a starting data macro, launch Microsoft Dynamics GP and begin recording the macro from the login window. Use the keyboard shortcut ALT-F8 to begin recording the macro. Record all the steps for the application setup and data entry that will be needed for the subsequent tests.

Setting up the Microsoft Dynamics GP Environment

Before recording test case macros, some setup is required in the Microsoft Dynamics GP environment.

Macro Folder

Create a folder within the Data folder of the Microsoft Dynamics GP installation that has the same name as the test case (the macro name). This folder is used to store the macro file, any data dumps, and any report output that is generated during recording.

Applications

Be sure that you have installed the required applications for the Microsoft Dynamics GP installation. Be sure to install any online help files if these will be tested in your test case.

If you are using any external software, such as automated testing tools or bug tracking systems, be sure those are set up as well.

System Settings

If the test cases you are recording involve printing, set the default printer for the system. Turn off any screen savers and any other applications that could interfere with Microsoft Dynamics GP. This is most important during automated regression tests, when you are running batches of macros.

Dex.ini File

Several settings in the Dex.ini file control the macro system. The Dex.ini file is typically found in the Data folder of the Microsoft Dynamics GP installation.

Report Output Location

Add the ReportPath setting to the Dex.ini file to specify the base location where the report output will be saved. Typically, this path will be the Data folder of the Microsoft Dynamics GP installation. When reports are exported to text files, the macro will record the path relative to this location. The ReportPath setting has the following syntax:

```
ReportPath=C:\Program Files\Microsoft Dynamics\GP\Data
```

Advanced Macro Menu

The Advanced macro menu must be available. You will need to use commands from this menu when working with macros. Make the menu available by adding the following entry to the Dex.ini file:

```
ShowAdvancedMacroMenu=TRUE
```

Date, Time, and Currency Formatting

It's important to have consistent formatting for dates, times, and currency values when recording and playing back test case macros. This reduces the number of differences reported when the test case is re-run during regression testing. The following settings in the Dex.ini file are used to ensure consistent date formatting.

Setting	Description
StdInternationalInfo	This setting allows the operating system settings for date, time, and currency formatting to be overridden. TRUE = Use Dexterity standard formats for date, time, and currency values FALSE = Use control panel settings for date, time, and currency values
StdInternationalInfoDateCentury	This setting allows control over the year portion of the date values used by StdInternationalInfo. TRUE = Four digits will be used for the year portion of date values FALSE = Two digits will be used for the year portion of date values
StdInternationalInfoDateZeroPrefix	This setting allows control over the date format for SdtInternationalInfo. TRUE = Single-digit day and month values will be prefixed with a 0 (zero) FALSE = Single-digit day and month values will not be prefixed

Keys.ini File

The Keys.ini file contains macro commands that are run at startup to configure various system behaviors. The macro commands in the Keys.ini are run only if the file is included in the command to launch Microsoft Dynamics GP. You create this file using a text editor such as Notepad. The Keys.ini file is typically stored in the Data folder of the Microsoft Dynamics GP installation. **To work properly, the Keys.ini file must also have a blank line as the last item in the file.** The following macro commands are typically used in the Keys.ini file. You could also use these commands at the beginning of any macro file.

Static System Date and Time

We recommend that you use a static system date and time when recording and running macros for test cases. This helps reduce report differences when the macros are re-run. The Keys.ini file can be used to set the date and time values so they are static. The following macro commands are used to ensure static date and time values:

Setting	Description
SetEbDateTo	This command specifies the system date value to use. The possible values are: 'o/s' = Use the operating system date date MM/DD/YY/DOW = Use the date specified, where MM is the month, DD is the date, YY is the year, and DOW is the day of the week Example: SetEbDateTo date 01/01/1965/3
SetEbTimeTo	This command specifies the system time value to use. The possible values are: 'o/s' = Use the operating system time time HH:MM:SS = Use the time specified, where HH is the hour (from 0 to 23), MM is the minute, and SS is the second Example: SetEbTimeTo time 4:15:00

Application Behavior

Commands in the Keys.ini file control how the Microsoft Dynamics GP application behaves when macros are recorded and run. The following macro commands control application behavior:

Setting	Description
Logging	This command writes macro-related warnings and messages to a file, rather than to the screen. The command has the following syntax: Logging file filename Filename = The name of the file that the macro messages will be stored in. This name can include a generic path. Example: Logging file 'regression_log.txt'
SetExitAction	This command specifies the behavior when a macro crashes. The possible values are: 'o/s' = Close the application upon macro crash 'nothing' = Leave the application open upon macro crash Example: SetExitAction 'o/s'
MacroError	This command specifies the number of macro errors that must be encountered before the exit action is performed. The command has the following syntax: MacroError limit value value = An integer that specifies the number of errors Example: MacroError limit 5

Setting	Description
FilePath	<p>This command specifies whether file paths that are stored in the macro will be absolute or relative. In most cases, relative paths should be used, because they are more likely to work when test case macros are played back on other Dynamics GP installations. The possible values are:</p> <p>relative = Use relative paths absolute = Use absolute paths Example: FilePath relative</p>

The following is an example of a Keys.ini file:

```
SetEbDateTo date 01/01/1965/3
SetEbTimeTo 'o/s'
Logging file 'Regression_Test.log'
SetExitAction 'o/s'
MacroError limit 5
FilePath relative
```

Recording a Test Case

Tests should be designed with the intent of being recorded using the Dexterity macro system. In cases where the code has too many problems to complete the initial recording, the macro script can be deleted and a new recording started after the code becomes more stable. The person recording the test case macro will need to decide whether the defects in the code will cause significant rework in the macro, or whether the edits will be fairly simple. If the edits required are significant, wait until the code is more stable before recording the macro.

Review the test case before you attempt to record it. Review any bug reports that are associated with the test case before you record it. Use that information to determine how to best incorporate them into a test case. The bug reports may also indicate areas where a “workaround” will be necessary to complete the automation of the test case.

To record a test case, use the following procedure:

1. Launch Microsoft Dynamics GP. If you are using a separate testing tool that supports it, use that tool to launch Microsoft Dynamics GP. Be sure that you are referencing the Dex.ini and Keys.ini files that you modified earlier. The easiest way to include them is by launching from the command line, as shown in the following example:

```
C:\Program Files\Microsoft Dynamics\GP\Dynamics.exe Dynamics.set "C:\Program Files\Microsoft Dynamics\GP\Data\Dex.ini" "C:\Program Files\Microsoft Dynamics\GP\Data\Keys.ini"
```

2. At the login window, use the ALT-F8 keyboard shortcut to begin recording a macro. Type the name of the macro. Set the location where the macro is being saved to the folder you created in the Data folder of the Microsoft Dynamics GP installation. Click Save to begin recording the macro. All actions from this point forward will be recorded.
3. Log in to Microsoft Dynamics GP. Be sure to enter the login name and password explicitly, rather than accepting any default values. This ensures the values are recording in the macro. If test case macros will be re-run on different systems, do not record the selection of the SQL Server in the login window. Instead, use the default value for the current Microsoft Dynamics GP installation.
4. Open the Insert Macro Header window. Choose Microsoft Dynamics GP >> Tools >> Macro >> Advanced >> Insert Header. The macro header provides information such as who recorded the macro, details about starting data, and what tests are being performed. This information will be useful for someone who may need to edit the macro in the future. Enter the following items:

Keystroke - Macro number/name of the test

Module – A code that indicates the area the test is associated with

Version Number – The product version the macro was created for

Created by – The login of user for the test or the alias of creator

Beginning Data – The starting data test case number and/or related modules

Description – Details about the test being created

Problem reports – A list of the problem reports or bug fixes associated with the test case

Click OK to save the macro header.

5. Begin recording the steps of the test case. Refer to the section [Tips When Recording Test Cases](#) for hints that will make recording the test cases easier.
6. Use the window, field, menu, and table dump capabilities to create snapshots of the state of the application. You will create a baseline version of these files, and use them for comparison with the dump files created when you run the test macro during a regression. The following table lists the types of dumps available when recording macros:

Type	Description
Window Dump	Creates a bitmap graphic image of the window
Field Dump (All)	Creates a text file that contains information about the fields on a window. The information in a field dump includes the following: <ul style="list-style-type: none">• Window Name• Form Name• Visible fields in the window, including data displayed• Details about fields, such as static text values for list boxes, and accelerator keys for push buttons• Status of fields, such as being locked or disabled• Shrink or expand status of scrolling windows
Field Dump (Selected)	Creates a text file that contains information about a specified set of fields on the window. When selecting fields, include a key indicator field on the window such as the Item Number or Customer Name.
Menu Dump	Creates a text file that contains information about the menus for a window. Only the form-level and form trigger menus are included. The core menus are not included.
Menu Dump (All)	Creates a text file that contains information about the menus for a window. All menus are included.
Toolbar Dump All	Creates a text file that contains information about all toolbars in the application.
Background Dump	Creates an XML file that contains the code used for the background currently displayed in the main Microsoft Dynamics GP window. Use this command when testing the Home page or area pages.
Shell Dump	Creates an XML file that contains information about the shell (main window) for Microsoft Dynamics GP. Use this command when testing the navigation pane.
Table Dump	Creates a text file that contains the specified columns from a table. Tables contain some columns (like the note index) that can be different every time the test case macro is run. These columns should be unmarked so they do not create unnecessary differences when comparing the results.

7. If the test case contains reports, print each report to the File destination. Always use the path lookup button when specifying the location where the report will be exported. Do not type the report path. Typically, you will choose to replace the existing report. If you choose the Append option, the report data will be appended to the output each time the test case macro is run.

If you need to verify actual printed copies of a report for report layout testing, it is best to verify them in a separate printer-specific test.

8. To end the recording, first close all windows in Microsoft Dynamics GP. To close the application, choose Microsoft Dynamics GP >> Exit.

Tips When Recording Test Cases

Use the following tips when recording test cases.

Comments

Comments can be added to macros by choosing Insert Comment from the Advanced macro menu. The comments will be added to the macro, but have no effect when the macros are run. All comment lines in the macro begin with the pound sign (#). They are used to provide detailed information when the macros are viewed or edited. Exercise discretion when deciding the quantity and location of the comments you add to test case macros. Comments are used in the following cases:

- To describe the test case, the areas being tested, and the expected results.
- To describe the areas of the test case that are complex or vague.
- To indicate what item was selected from a lookup window or scrolling window.
- To describe work-arounds that are implemented in the macro.

Fields

Enter data into fields directly instead of using a lookup. This can significantly reduce maintenance of macros as the test data changes. If you are testing lookups, select the data from the lookup and add a comment following the selection noting in the macro which selection was made from the lookup.

Windows

Do not move windows while recording a test case macro. Several lines will be added to the macro for every pixel the window is moved. This adds clutter to the macro and makes editing the macro difficult.

Do not minimize windows while recording a test case macro. The minimize action is recorded as a window move, and adds several lines to the macro.

Be aware of the system screen resolution when you record macros. If macros are played back at a different screen resolution, windows may open at a different size. This can affect macro results. For example, at a lower screen resolution, a scrolling window may display fewer rows.

If a window is not accessible through the standard navigation, you can use the Open Form command from the Advanced macro menu to open it.

Scrolling Windows

Do not use the scroll box to navigate the content of the scrolling window. Use the page up and page down keys on the keyboard, or click the arrows on the scroll bar. Using the scroll box records many extra lines in the macro that are difficult to trace and determine where the macro is trying to focus.

Field Dumps

Before you record a field dump, be sure to move the focus off of the last field you entered data into. A field dump will not record data unless the focus has moved off of the field. When possible, use the selected field dump to have better control over what data is included in the field dump. Always include the key fields on the window.

Reports

Do not print reports to the screen or printer. Reports sent to the screen or printer cannot be verified when the test case macro is re-run.

When recording report dumps, name them in sequential order such as .r001, .r002. This helps with verification and maintenance because it is easier to know the order in which the report dumps were created. If you know there are additional report dumps to be added to the test case, leave the appropriate amount of report numbers available when recording.

Application Problems When Recording Test Cases

You may encounter application problems when you try to record test cases with the macro system. In some cases, you can work around these problems and record an initial version of the test. If a test case cannot be completely recorded, consider adding a "crash line" to the macro. This is simply a line of text within the macro that cannot be processed and will cause the macro to crash. For instance, adding the line "Crash Here" in the macro will cause it to crash. It's a good idea to add a crash line to an incomplete macro so it won't accidentally be considered complete. When the test case macro can be completed, remove the crash line.

Incomplete Functionality

The application for which you are recording test cases for may not be complete at the time when you record them. In this case, you can record as much of the test case as possible, up to the steps where the functionality does not exist. Once the functionality is available, you can record the remainder of the test case. Be sure to fully document what is left to record in the test case, and whether any macro edits will be required when the functionality is complete.

Bugs in the Application

You may encounter application bugs that prevent the test case from being completed. In some cases, you can use a work-around for the bug to continue recording the test case. Be sure to add a comment and a "crash line" to the macro so the macro can be fixed when the application bug is resolved. If the work-around requires extensive editing of the macro, it might be better to wait until the bug in the application is fixed before recording that section of the test case.

Running a Regression

In a regression test, you will run all of your test case macros to verify a build of your application. You can run the test case macros manually, or you can use an automated method to run them. The simplest automated method is a batch (.bat) file. The batch file can set the working folder, clear the macro log file, and then run the test case macros. To run a macro from a batch file, use this syntax:

```
Dynamics.exe Launch_file Dex_ini_file Keys_ini_file Macro_file
```

If you are not using a Keys.ini file, exclude it from the command.

The following example shows a batch file that runs several test cases for a regression.

```
REM --- Set the current folder and delete the regression test log ---
C:\
cd C:\Program Files\Microsoft Dynamics\GP\
del .\Data\Regression_Test.log
REM --- Run the test case macros ---
Dynamics.exe Dynamics.set .\Data\Dex.ini .\Data\Keys.ini .\Data\Test\LeadMaintTest.mac
Dynamics.exe Dynamics.set .\Data\Dex.ini .\Data\Keys.ini .\Data\Test\LeadInquiryTest.mac
Dynamics.exe Dynamics.set .\Data\Dex.ini .\Data\Keys.ini .\Data\Test\LeadLookupTest.mac
REM --- Pause at the end ---
pause
```

Verifying the Test Case Results

After you have run the test case macros, you will compare the results to the baseline that you created. First, you will determine whether all of the test case macros completed successfully. Do this by searching the macro log file for any errors that are logged. If any of the test case macros did not finish, investigate and identify the cause of the problem.

Next, you will compare the various output files created by each test case macro to those from the baseline versions for each test case. Use a text comparison tool such as WinDiff to compare the files. If the files generated by the test case do not match those from the baseline exactly, you must investigate to find the cause of the difference. Be sure to document all of the differences you find. When the application problem is resolved, you can use this documentation to verify that the differences have been resolved.

When verifying the reports generated by test cases, it is important to scan the entire report for any noticeable defects. You should not rely on only the differences between the expected baseline results and the results generated by the test cases.

Maintaining Test Case Macros

Test case macros require maintenance just like any other code for your application. Use the following guidelines when maintaining your test case macros:

- Keep your test case macros and baseline test results in a safe location, such as a source code repository.
- Plan to have a separate set of test case macros for each major version of Microsoft Dynamics GP. When creating the test cases for a new major version, you can start with a copy of the test cases from the previous major version.
- Expect to edit the test case macros so that they will continue to work when changes are made in Microsoft Dynamics GP or your integrating application.

Additional Resources

Refer to the Dexterity help file (Dex.chm) for detailed information about the windows in the Macro System. The help file also contains descriptions of the commands in the macro language.

Conclusion

By using the techniques described in this document, you can record test case macros to include in a set of regression tests for your integrating application. These tests can help you identify problems that occur when you make changes to or upgrade your application to work with new releases of Microsoft Dynamics GP.

Microsoft Dynamics is a line of integrated, adaptable business management solutions that enables you and your people to make business decisions with greater confidence. Microsoft Dynamics works like and with familiar Microsoft software, automating and streamlining financial, customer relationship and supply chain processes in a way that helps you drive business success.

U.S. and Canada Toll Free 1-888-477-7989

Worldwide +1-701-281-6500

www.microsoft.com/dynamics

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, this document should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2008 Microsoft Corporation. All rights reserved.

Microsoft, the Microsoft Dynamics Logo, Microsoft Dynamics, and Windows are either registered trademarks or trademarks of Microsoft Corporation, or Microsoft Business Solutions ApS in the United States and/or other countries. Microsoft Business Solutions ApS is a subsidiary of Microsoft Corporation.

Microsoft